

poems

La revue francophone des utilisateurs de l'Apple

- 🍏 **Hard-copy HGR sur ImageWriter**
- 🍏 **Macintosh : deux accessoires**
- 🍏 **Disquette mixte Dos/Pascal**
- 🍏 **Un analyseur de syntaxe**
- 🍏 **Intelligence artificielle**
- 🍏 **Catalogue Multi-Sed**
- 🍏 **Catalogue Souris**
- 🍏 **Méthode PERT**

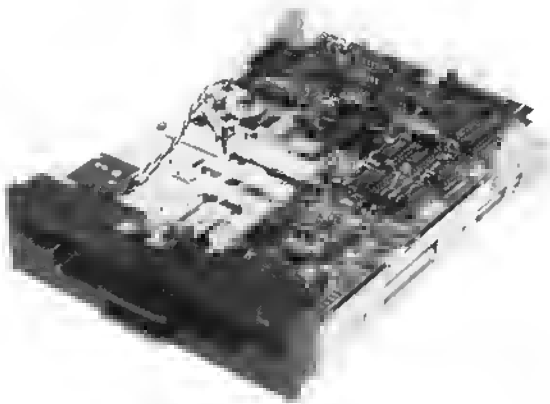
NUMERO 19 • PRIX 40 F

ISSN : 0294-6068

DRIVE CHINON

5"1/4 COMPATIBLE APPLE II+, IIe, IIc

NOUVEAU: Câble pour IIc



- 1/2 HAUTEUR
- 40 PISTES (48 T.P.I.)
- 143 Ko SOUS DOS 3.3
- 160 Ko SOUS DOS MODIFIE
- DETECTION DE PISTE 0 ET PROTECTION
- ECRITURE PAR OPTO-ELECTRONIQUE
- MOTEUR A ENTRAÎNEMENT DIRECT
- SYSTEME DE FERMETURE BREVETE

LE MEILLEUR RAPPORT QUALITE/PRIX DU MARCHÉ

PROMOTION

Floppy simple 1 256 Frs H.T 1 490 Frs T.T.C
Floppy double 2 521 Frs H.T 2 990 Frs T.T.C
(prix par quantité sur demande)

*Ce lecteur de disquette a été homologué par la FRAMIF,
l'Aéropotiole, le club Thomson, etc...*

- HAUTE FIABILITE
- SILENCIEUX
- FAIBLE CONSOMMATION
- GARANTIE 1 AN



CE MODELE FAIT PARTIE D'UNE GAMME COMPLETE DE LECTEURS 5"1/4 & 3"1/2 AVEC INTERFACE SASI.
Autres modèles disponibles pour tous micros (nous contacter)

PROMO | - Carte 80 colonnes étendue 990 T.T.C
- Disquettes 5"1/4 11,90 Frs H.T par boîte de 10. (13,90 Frs T.T.C)
- Moniteurs 12" haute résolution vert ou orange 990 Frs T.T.C

MICRO-DISPO
58, rue Blomet

Tél.: 566 57 17
75015 PARIS

P.I.E.D
42, bd Magenta

Tél.: 249 16 50
75010 PARIS

NOM: PRENOM: Tél.:

ADRESSE:

SOCIETE: PROFESSION:

Veuillez m'envoyer une documentation sur vos produits.

Je suis intéressé à titre personnel..... ☐ professionnel ☐

Editorial



Hervé Thiriez

5

Un analyseur de syntaxe



Carlos Sacré

6

Un catalogue Multi-Sed



Alain Gogniat

17

Une mémoire-tampon d'écran



Gérard Michel

29

Lucy in the Sky with Diamonds



Jean-Luc Bazanegue

42

Disquettes mixtes DOS/Pascal



François Sermier

63

Mousecat



Patrice Neveu

68

Erratum



61

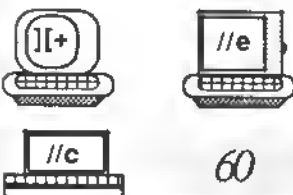
La méthode PERT



Serge Cattan

23

Retrouvez vos programmes perdus



Guy d'Herbement

60

Création de caractères



Jean-Pierre Vinter

52

Où est la souris ?



Marianne Sutz

40

Bibliographie



Alexandre Duback

69

Résultat du sondage



75

Courrier des Lecteurs



Olivier Herz

72

Etrange accessoire



Evelyn Gorin

50

Fermez les fenêtres !



Julien Thomas

50

Hard-copy HGR sur ImageWriter



Yvan Koenig

59

L'intelligence artificielle



Guido Bettiol

35

Le système de développement 68000



Cécile Derrien

51

Les annonceurs

Apple : pages 38 et 39 / Belden GmbH : page 75 / IEF : page 2
List : page 76 / Télcompa : page 4

Edition MEV

64/70, rue des Chantiers - 78000 Versailles

Directeur de la publication : Hervé Thiriez

Micro informations

Jean-Michel Gourévitch



70

photocomposition :

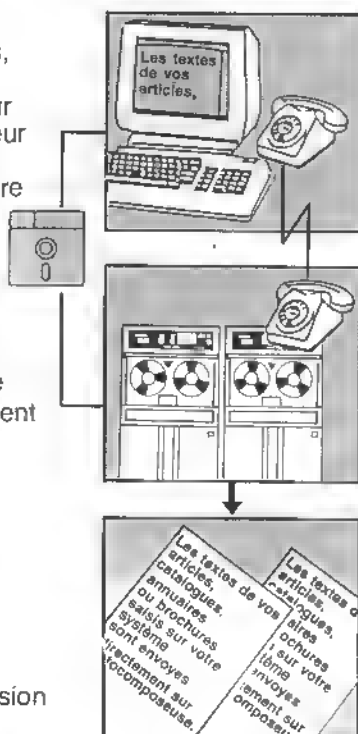
TELECOMPO

c'est facile
c'est économique
et ça va vite !

Les textes de vos articles, catalogues, annuaires ou brochures saisis sur votre micro-ordinateur sont envoyés directement sur notre photocomposeuse.

Nous vous évitons ainsi, le coût et le temps de la saisie supplémentaire que nécessite le traitement traditionnel de la photocomposition avant l'impression des documents.

Si vous le désirez nous pouvons également nous charger de l'impression et du brochage.



TELECOMPO (1) 328.18.63

13 et 15, avenue du Petit-Parc - 94300 VINCENNES

PHOTOCOMPOSITION - BUREAUTIQUE -
TRANSMISSION DE DONNÉES -
GESTION DE FICHIERS -
MATÉRIELS DE TRAITEMENT DE TEXTES

Une référence : la revue
pom's

Basicium...

**le Basic
enrichi**

**Gérard
Michel**

**Apple
II+, IIe**

**Cet utilitaire vous
permet de :**

**Gérer des masques
40 ou 80
colonnes. Ces
écrans, définis lors
de la
programmation,
s'affichent par la
simple commande
"JA".**

**Imprimer
l'écran 40 ou
80 colonnes
par "JH".**

**Nettoyer
mémoire très
rapidement
par "JF".**

**Saisir toutes les
variables d'un
écran en une seule
commande "JI" en
contrôlant le type
et la longueur.**

**Gérer des
messages, la
réponse étant
contrôlée.**

**Mais aussi, ne
saisir que l'une
des variables
affectées à un
écran, les afficher,
les effacer, etc...**

**Tous les fichiers source sont sur la disquette
(exemple de démonstration dans le Pom's 13)**

**Disquette et documentation :
150,00 TTC franco
(bon de commande page 74)**

Apple ferme trois usines et licencie 1 200 personnes. Est-ce la fin de notre beau rêve ? Certes non car, dans la foulée, nous apprenons aussi les informations suivantes :

- IBM, après avoir abandonné le PC Junior, annonce une forte baisse de ses bénéfices;
- Hewlett Packard pratique du chômage technique sur 4 500 salariés;
- Xerox a de grosses difficultés;
- Wang licencie 1 500 personnes;
- Data General licencie 1 300 personnes;
- Control Data et Honeywell ferment des unités;
- Atari annonce trois mois de retard sur le "Jarkintosh", le futur concurrent du Macintosh;
- enfin, on ne compte plus les distributeurs qui, quelle que soit leur taille, déposent leur bilan ou deviennent soudainement mauvais payeurs.

Est-ce à dire que la vague de la micro-informatique est définitivement enrayée ? Certes pas ! Selon l'avis de la plupart des professionnels, qui ne se trompent quand même pas tout le temps, il s'agit là d'une période de consolidation. Une fois que tout sera rentré dans l'ordre, la micro-informatique devrait retrouver un rythme de croissance qui, entre 15% et 20% par an, sera plus calme que ce que nous avons vécu ces dernières années, mais beaucoup moins vulnérable.

En tous cas, la confiance en Apple ne manque pas puisqu'il y a maintenant trois revues entièrement consacrées au Macintosh, avec Infomac, Mic Mac et, plus récemment, Icônes. Ceci dit, Pom's reste la seule revue au monde où les amateurs de Mac trouveront, outre des informations, des programmes permettant de s'immiscer dans le Mac et de réellement comprendre et exploiter son mode de fonctionnement. Les amateurs d'Apple II, quant à eux, en auront aussi pour leur argent : fidèle à ses promesses, Pom's maintient le nombre de pages consacrées à la famille du 6502.

Nous vous remercions tous de votre réponse à Apple Expo, où le seul problème rencontré à notre stand était parfois le trop grand nombre de lecteurs venus se réabonner ou tout simplement pour nous dire bonjour. Rendez-vous à la boutique Sicob (Porte Maillot) où nous vous recevrons au stand M279.

Chasse promise dans le précédent numéro, chose due : dans ces pages vous trouverez l'analyseur de syntaxe par disquettes entières de Carlos Sacré.

Sans entrer dans le détail du sommaire, plusieurs titres sont à souligner :

Au chapitre "Operating System", le catalogue Multi-Sed d'Alain Gogniat et la disquette mixte Dos/Pascal de François Sermier.

Au chapitre initiation, l'approche de l'intelligence artificielle de Guido Bettiol, la méthode PERT en Basic de Serge Cattani et, toujours pédagogique, l'utilisation de la carte langage par Gérard Michel.

Au chapitre initiés, le catalogue à l'aide de la souris de Patrice Neveu dont le source trop volumineux n'a pu être listé dans la revue.

Au chapitre utilitaires, un patch d'Yvan Koenig pour un hard-copy de la page graphique haute résolution et la création de caractères programmables sur l'ImageWriter de Jean-Pierre Vinter.

Au chapitre Macintosh, un nouvel accessoire de bureau Lucy in the Sky with Diamonds (!) dû à Jean-Luc Bazanegue dont le source est impressionnant; un accessoire clin-d'œil d'Evelyne Gorin et deux utilitaires indispensables de Julien Thomas et Marianne Sutz.

Enfin, au chapitre offre d'emploi, nous sommes toujours à la recherche d'un collaborateur à temps plein ayant des connaissances très élevées au niveau de l'assembleur 6502, ou au moins d'un collaborateur à mi-temps maîtrisant parfaitement l'assembleur, le IIc et le ProDOS.

Bonnes vacances !

Hervé Thiriez

Ont collaboré à ce numéro : Jean-Luc Bazanegue, Guido Bettiol, Serge Cattani, Cécile Derrien, Alexandre Duback, Alain Gogniat, Evelyne Gorin, Jean-Michel Gourévitch, Guy d'Herbemont, Olivier Herz, Yvan Koenig, Gérard Michel, Patrice Neveu, Carlos Sacré, François Sermier, Marianne Sutz, Julien Thomas, Jean-Pierre Vinter.

Directeur de la publication, rédacteur en chef : Hervé Thiriez.

Dessins : Laurent Bidot.

Rédacteurs : Alexandre Avrane, Olivier Herz.

Siège social : Editions MEV - 49, rue Lamartine - 78000 Versailles - Tél. : (3) 951.24.43.

Publicité : Consulter les Editions MEV.

Diffusion : N.M.P.P.

Composition : Télécompo - 13/15, avenue du Petit Parc - 94300 Vincennes - Tél. : 328.18.63.

Impression : Rosay - 47, avenue de Paris - 94300 Vincennes - Tél. : 328.18.63.

Un analyseur de syntaxe en langage machine

Carlos Sacré

Le but de ce logiciel est la détection des erreurs de syntaxe dans les programmes écrits en Basic APPLESOFT. La syntaxe est la technique d'écriture des divers symboles composant une instruction, de façon que l'interpréteur Basic puisse la comprendre. Le remplacement d'un "point-virgule" par un "deux-points", l'oubli d'une parenthèse, la substitution d'une variable numérique A à une variable alphanumérique A\$, sont des erreurs de syntaxe. Les erreurs de logique (AND au lieu de OR, branchement au moyen d'un GOTO à l'intérieur d'une boucle FOR... NEXT... etc...) sont d'un type tout à fait différent, et sortent du cadre de cet article.

Tout programmeur, même novice, sait qu'il ne suffit pas qu'un programme d'une certaine ampleur ait fonctionné une ou plusieurs fois pour qu'il soit exempt de toute erreur : une utilisation qui fait intervenir une ligne rarement appelée peut en effet apporter une interruption surprise, désagréable au beau milieu d'une partie intéressante, ou d'un programme de gestion. Chacun sait aussi qu'il est particulièrement facile d'introduire une "bête erreur" quand on met au point ou que l'on modifie un programme. D'où l'intérêt d'avoir un logiciel permettant de vérifier rapidement la parfaite correction de ses programmes.

Le système présenté ci-dessous est adapté aussi bien à une vérification fréquente en cours de mise au point, qu'au contrôle systématique en une seule opération de tous les programmes Basic APPLESOFT d'une face de disquette.

Mode d'emploi du système

a) Vérification d'un programme isolé, déjà résident en mémoire vive. Il suffit de commander "BRUN SNTX" après avoir placé dans le lecteur voulu une disquette avec ce programme binaire. Des appels ultérieurs nécessités par la mise au point du programme Basic peuvent se faire simplement par "CALL 35072" ou "&", tout au moins tant que HIMEM; ou l'adresse de & n'ont pas été modifiées (par exemple par un essai intermédiaire du programme Basic si celui-ci modifie cet environnement).

b) Vérification d'un ou plusieurs programmes implantés sur une disquette. Placer dans le lecteur 1 une disquette non protégée en écriture, comportant SNTX, SYNTAXE, SYNAXE-TEST, FINSYNTAXE, et commander "RUN SYNTAXE, D1".

On a alors la possibilité d'obtenir des renseignements sur le fonctionnement du système, puis on est invité à placer dans le drive 2 la disquette B à tester, et enfin à choisir une option : soit tester un seul programme, soit tester tous les programmes de la disquette. Après le test, on peut recommencer avec une nouvelle disquette B, ou reprendre la main : dans ce dernier cas, les pointeurs du DOS sont replacés dans leur état normal par la commande "FP".

NDLR : dans le cas b), il est préférable d'avoir dans le lecteur 1 une copie de sauvegarde et non un original, car l'opération peut parfois, pour des raisons (dues au DOS) qui nous échappent, "planter" le système. Il est conseillé de protéger contre l'écriture la disquette du lecteur 2.

Mode de fonctionnement du système

Programme (binaire) SNTX :

C'est le coeur du système : écrit en langage machine grâce à l'assembleur LISA 2.5, c'est lui qui effectue le test proprement dit. Il décode ligne par ligne, instruction par instruction, octet par octet, tout le programme Basic résident en mémoire vive. Il affiche à l'écran le numéro de ligne, le rang de l'instruction dans la ligne, et une brève appréciation telle que "CORRECT", "VIDE" (ligne réduite à un ; ou :: dans une ligne), "INSTRUCTION REM"... Toutes les commandes du Basic APPLESOFT sont étudiées en détail, sauf "USR", "CALL", "&" (qui sont parfois suivies de paramètres que SNTX ne pourrait interpréter), "DATA" (c'est au concepteur à ne pas mélanger les types de données) et "GET" numérique (GET A est déconseillé : il faut utiliser GET A\$), pour lesquelles l'instruction est listée. Il appartient à l'utilisateur de vérifier la syntaxe de ces instructions, selon leur rôle dans le programme. De plus, SNTX contrôle l'existence dans le programme de toute ligne susceptible d'être appelée par un "GOTO".

"GOSUB", "ON GOTO", "ON GOSUB" ou "ONERR GOTO".

Le défilement à l'écran est très rapide (plus que pour un listing !), mais il est toujours possible de le suspendre, grâce à Ctrl-S, pour une étude approfondie des messages. Dès qu'une erreur est décelée, le test est provisoirement interrompu avec émission d'un "bip", l'instruction incriminée est listée, avec les éventuels caractères de contrôle en affichage clignotant, et le type de l'erreur est signalé par un bref message. L'utilisateur peut donc noter le renseignement souhaité, puis commander la reprise du test en enfonçant n'importe quelle touche.

Certains défauts d'écriture tolérés à l'exécution par l'interpréteur Basic sont relevés ; par exemple, l'omission des guillemets terminant une chaîne dans une instruction "PRINT" (c'est dangereux quand il ne s'agit pas de la dernière instruction de la ligne !). SNTX est implanté à l'adresse \$8900 (en décimal 35072), ce qui lui permet d'étudier des programmes Basic ayant jusqu'à 130 secteurs. Il place HIMEM; à cette adresse, ainsi que le vecteur d'appel de la commande ampersand (&). SNTX étant très court (13 secteurs), il est utile de le copier sur de nombreuses disquettes, en particulier celles qui servent pour les travaux courants.

Programme (Basic) SYNTAXE

C'est le programme qui permet de lancer la vérification automatique de tous les programmes Basic d'une face de disquette, sans intervention, sauf bien sûr pour relever les éventuelles erreurs. Il initialise le système, charge SNTX, renseigne l'utilisateur et lance SYNTAXE-TEST.

Programme (Basic) SYNTAXE-TEST

Il crée sur la disquette A (lecteur 1, avec le système) un fichier SNTX.EXEC établi à partir du catalogue de tous les programmes Basic APPLESOFT de la disquette B (lecteur2), ou d'un seul si on a choisi cette option, en négligeant tous les autres fichiers écrits sur cette disquette. SYNTAXE-TEST commande enfin l'exécution de SNTX.EXEC.

Fichier (texte) SNTX.EXEC

Il traite successivement tous les programmes voulus (chargés depuis B,

puis testés), automatiquement sauf en cas d'erreur : interruption puis redémarrage manuel comme pour l'utilisation au coup par coup. SNTX.EXEC passe ensuite la main à FINSYNTAXE.

Programme (Basic) FINSYNTAXE

Il efface de la disquette A le fichier SNTX.EXEC, puis, suivant le choix, rend la main après avoir commandé

"FP", ou relance SYNTAXETEST pour l'exploration d'une autre face de disquette.

Comment faire avec un seul lecteur ?

Le système de contrôle automatique est normalement prévu pour deux drives; il est possible de l'adapter pour un seul : modification des données RWTS dans le programme SYNTAXETEST (de façon à faire lire le catalogue dans le lecteur 1), inter-

ruptions pour demander le changement de disquette, avant d'écrire SNTX.EXEC, et après la fin du test. Toutefois, il ne pourra alors tester que des disquettes non protégées en écriture et comportant suffisamment de place disponible pour le fichier SNTX.EXEC (environ un secteur pour deux programmes Basic sur la disquette).

Programme SYNTAXETEST

```

10 D$ = CHR$(13) + CHR$(4)
20 DIM A$(104)
30 TEXT : HOME
35 VTAB 5: HTAB 9: PRINT "ENTREZ LES DISQUETTES
:"
40 VTAB 9: PRINT "DRIVE 1 : SYSTEME DE DETECTION
D'ERREURS DE SYNTAXE."
50 PRINT : PRINT "DRIVE 2 : DISQUETTE AVEC
LE OU LES PRO- GRAMMES A TESTER."
60 VTAB 23: HTAB 20: GET A$
70 HOME : VTAB 10: PRINT "1 TEST D'UN SEUL
PROGRAMME"
80 PRINT : PRINT : PRINT "2 TEST DE TOUS LES
PROGRAMMES APPLE- SOFT DE LA
DISQUETTE"
90 PRINT : PRINT : PRINT "ESC POUR ANNULER"
100 PRINT : PRINT : HTAB 2
110 GET A$: IF A$ < > "1" AND A$ < > "2" AND A$ < >
CHR$(27) THEN 110
120 IF A$ = CHR$(27) THEN 10000
500 FOR I = 768 TO 796: READ X: POKE I,X: NEXT I
530 PRINT D$"OPEN SNTX.EXEC,D1"
540 IF A$ = "2" THEN 2000
1000 HOME : VTAB 10: REM UN SEUL PROGRAMME
1010 PRINT "NOM DU FICHIER BASIC A TESTER :":
PRINT
1020 INPUT "":F$
1030 PRINT D$"WRITE SNTX.EXEC"
1040 PRINT "TEXT:HOME:PRINT" CHR$(34)F$ CHR$(
34)
1050 PRINT "POKE34,2"
1060 PRINT "LOAD "F$",D2"
1070 PRINT "CALL 35072"
1090 GOTO 3000
2000 HOME : VTAB 12: HTAB 5: PRINT "CREATION D'UN
FICHIER CATALOGUE."
2010 POKE 34,20: VTAB 22
2020 FOR SE = 15 TO 1 STEP - 1: REM TEST
DISQUETTE
2030 POKE 781,SE: CALL 768
2040 FOR F = 0 TO 6:FI = 105 - 7 * SE + F:A$(FI) = ""
2050 AD = 149 * 256 + 11 + F * 35
2060 IF PEEK (AD) > 35 THEN 2100
2070 CO = PEEK (AD + 2): IF CO < > 2 AND CO < > 130
THEN 2100

```

```

2080 FOR I = 3 TO 32:A$(FI) = A$(FI) + CHR$( PEEK (AD
+ I)): NEXT I
2090 IF PEEK (AD + 33) > 130 THEN PRINT "FICHIER
"A$(FI): PRINT "TROP LONG; NE PEUT ETRE
TRAITE !!!": PRINT CHR$(7): GET A$:A$(FI) = ""
2100 NEXT F,SE
2110 PRINT D$"WRITE SNTX.EXEC"
2120 FOR FI = 0 TO 104
2130 IF A$(FI) = "" THEN 2180
2140 PRINT "TEXT:HOME:PRINT" CHR$(34)A$(FI) CHR$(
34)
2150 PRINT "POKE34,2"
2160 PRINT "LOAD "A$(FI)",D2"
2170 PRINT "CALL 35072"
2180 NEXT FI
3000 REM CLOTURE FICHIER
3010 PRINT "RUN FINSYNTAXE,D1"
3020 PRINT D$"CLOSE SNTX.EXEC"
3030 TEXT : HOME
3040 PRINT D$"EXEC SNTX.EXEC,D1"
3050 END
10000 TEXT : HOME
10010 PRINT D$"FP"
10020 END
60000 DATA 169,3,160,8,32,217,3,96: REM APPEL
RWTS
60010 DATA 1,96,2,0,17,15,25,3,0,149,0,0,1,0,0,96,1:
REM IOB
60020 DATA 0,1,239,216: REM DCT

```

Programme FINSYNTAXE

```

10 D$ = CHR$(13) + CHR$(4)
20 PRINT D$"DELETE SNTX.EXEC,D1"
30 PRINT CHR$(7) CHR$(7) CHR$(7)"ENCORE (O/N) ?
:"
40 GET A$: IF A$ < > "O" AND A$ < > "N" THEN 40
50 TEXT : HOME
60 IF A$ = "N" THEN 90
70 PRINT D$"RUN SYNTAXETEST,D1"
80 END
90 PRINT D$"FP"
100 END

```


Programme SYNTAXE

```

10 HIMEM: 35072
20 D$ = CHR$(13) + CHR$(4)
30 TEXT: HOME
40 VTAB 12: HTAB 6: PRINT "DETECTION D'ERREURS
  DE SYNTAXE"
50 PRINT: HTAB 5: PRINT "SYSTEME ELABORE PAR
  CARLOS SACRE"
60 PRINT D$ "BLOAD SNTX"
70 POKE 35116,96: REM PERMET L'UTILISATION DE
  SNTX COMME SOUS-PROGRAMME
80 VTAB 24: HTAB 20: GET A$
90 HOME: VTAB 12: PRINT "DESIREZ-VOUS DES
  INFORMATIONS (O/N) ? ";
100 GET A$: IF A$ < > "O" AND A$ < > "N" THEN 100
110 IF A$ = "N" THEN 290
120 HOME: VTAB 9
130 PRINT " CE SYSTEME PERMET DE DECELER,
  DANS": PRINT: PRINT "LES PROGRAMMES EN
  BASIC APPLESOFT, LES"
135 PRINT: PRINT "ERREURS DE SYNTAXE, AINSI OUE
  L'ADRESSA-": PRINT "GE PAR DES 'GOTO',
  'GOSUB', 'ONERR' A": PRINT: PRINT "UNE LIGNE
  INEXISTANTE ."
140 VTAB 24: HTAB 20: GET A$
150 HOME
160 PRINT " IL SIGNALE ET LISTE LES INSTRUC-":
  PRINT: PRINT "TIONS DANS LESQUELLES
  FIGURENT LA FONC-": PRINT: PRINT "TION 'USR'
  OU LES INSTRUCTIONS '&','': PRINT: PRINT
  "CALL', 'DATA'."

```

```

170 PRINT: PRINT " LA COMMANDE 'CTRL-S' PERMET
  DE": PRINT: PRINT "SUSPENDRE LE DEFILEMENT
  A L'ECRAN ."
180 PRINT: PRINT: PRINT: PRINT " LES ERREURS
  DE SYNTAXE DONNENT LIEU": PRINT "A UNE
  INTERRUPTION DU TEST, DE MANIERE": PRINT:
  PRINT "A PERMETTRE A L'UTILISATEUR DE NOTER
  LA": PRINT: PRINT "LIGNE DE L'INSTRUCTION
  INCORRECTE ."
190 VTAB 24: HTAB 20: GET A$
200 HOME: VTAB 6
210 PRINT " CERTAINS DEFAUTS TOLERES PAR
  L'IN-": PRINT: PRINT "TERPRETEUR SONT
  RELEVES, PAR EXEMPLE": PRINT: PRINT
  "L'OMISSION DES GUILLEMETS APRES UNE":
  PRINT: PRINT "CHAINE TERMINANT UNE LIGNE ."
220 PRINT: PRINT: PRINT
230 PRINT " LA TAILLE DES PROGRAMMES TESTES":
  PRINT: PRINT "EST LIMITEE A 130 SECTEURS ."
240 VTAB 24: HTAB 20: GET A$
250 HOME: VTAB 7
260 PRINT " LA SUITE DU DEROULEMENT DE CE
  PRO-": PRINT: PRINT "GRAMME NECESSITE 2
  DRIVES . IL EST": PRINT: PRINT "TOUTEFOIS
  POSSIBLE AVEC UN SEUL DRIVE DE": PRINT
  "TESTER UN PROGRAMME ISOLE RESIDANT EN"
270 PRINT: PRINT "MEMOIRE VIVE, PAR LA
  COMMANDE .": PRINT: PRINT "BRUN SNTX' ."
280 VTAB 24: HTAB 20: GET A$
290 PRINT D$ "RUN SYNTAXETEST"
300 END

```

Programme SYNTAXE.TEXT (Assembleur LISA 2.5)

```

1 ;*****
2 ;*
3 ;* DETECTION DES ERREURS DE *
4 ;*
5 ;* SYNTAXE EN BASIC APPLESOFT *
6 ;*
7 ;*****
8 ;
9 ORG $8900
10 ;
11 ;POINTE DANS LE PROGRAMME BASIC
12 PTR EPZ $06
13 NL EPZ $08
14 LN EPZ $18
15 ;
16 ;POINTEUR DE PILE
17 I EPZ $ID
18 DRGET EPZ $IE
19 INSTR EPZ $CE
20 ERR EPZ $CF
21 SUP EPZ $EB
22 ECA EPZ $EC
23 INF EPZ $ED
24 LNG EPZ $EE
25 LIN EPZ $FA
26 PILE EQU $300
27 SETINV EQU $FE80
28 SETNORM EQU $FE84
29 BELL EQU $FF3A
30 COUT1 EQU $FDF0
31 STROUT EQU $DB3A

```

```

32 HOME EQU $FC58
33 CR EQU $FC62
34 ;
35 PROGRAMM JSR HOME
36 ;
37 ;&, HIMEM:, POINTEUR
38 LDA #$4C
39 STA $3F5
40 LDA #$00
41 STA $73
42 STA $3F6
43 LDA #$89
44 STA $74
45 STA $3F7
46 LDA $67
47 STA PTR
48 LDA $68
49 STA PTR+1
50 ;
51 ;PASSAGE A LA LIGNE SUIVANTE
52 BOUCLE JSR INCPTR
53 STA NL
54 JSR INCPTR
55 STA NL+1
56 ORA NL
57 BNE >I
58 JMP $03D0
59 ^I JSR INCPTR
60 STA LN
61 JSR INCPTR
62 STA LN+1
63 JSR CODAGE
64 JSR SETINV
65 HEX A9
66 BYT TXLIGNE
67 HEX A0

```

```

68 HBY TXLIGNE
69 JSR STROUT
70 LDA #$01
71 STA INSTR
72 ;
73 ;PASSAGE INSTRUCTION SUIVANTE
74 NINSTR LDA $SOB
75 STA $24
76 JSR SETINV
77 LDA INSTR
78 STA LN
79 LDA #$00
80 STA LN+1
81 JSR CODAGE
82 HEX A9
83 BYT NUMLIN+3
84 HEX A0
85 HBY NUMLIN+3
86 JSR STROUT
87 JSR SETNORM
88 INC $24
89 LDX #$00
90 STX I
91 STX ERR
92 JSR TRAIINS
93 PHA
94 CPX #$07
95 BNE >1
96 LDY ERR
97 BEQ >I
98 LDX $SOB
99 ^1 TXA
100 AND $OF
101 PHA
102 CMP $01
103 BEQ >2

```


104	CMP #\$02	181	LDA PILE,Y	258 ^2	CMP #\$22	;"
105	BNE >3	182	STA PTR	259	BNE >3	
106 ^2	PHA	183	INY	260	PHA	
107	TXA	184	LDA PILE,Y	261	LDA ERR	
108	LSR	185	STA PTR+1	262	EOR #\$01	
109	LSR	186	RTS	263	STA ERR	
110	LSR	187 ;		264	PLA	
111	LSR	188 ;DIVISION PAR 10		265 ^3	CMP #\$B2	
112	ADC #\$0E	189 DIVISE LDX #\$10		266	BNE >4	
113	JSR MESSERR	190	LDA #\$00	267	LDX \$FF	
114	PLA	191	ASL LN	268	STX ERR	
115 ^3	JSR MESSERR	192	ROL LN+1	269 ^4	CMP #\$80	
116	PLA	193	ROL	270	BCC >5	
117	PHA	194 ^1	CMP #\$0A	271	SEC	
118	CMP #\$07	195	BCC >2	272	SBC #\$7F	
119	BEQ >4	196	SBC #\$0A	273	TAX	
120	CMP #\$08	197 ^2	ROL LN	274	LDA #\$20	
121	BEQ >4	198	ROL LN+1	275	JSR OUTDO	
122	CMP #\$0C	199	ROL	276	LDY #\$D0	
123	BEQ >4	200	DEX	277	STY \$9D	
124	JSR LISTING	201	BNE <1	278	LDY \$SCF	
125 ^4	PLA	202	ROR	279	STY \$9E	
126	CMP #\$07	203	RTS	280	LDY \$FFF	
127	BCS >6	204 ;		281 ^6	DEX	
128	JSR BELL	205 ;TRADUCTION HEXA --> DECIMAL		282	BEQ >8	
129	STA \$C010	206 CODAGE LDY #\$04		283 ^7	JSR \$D72C	
130 ^5	LDA \$C000	207 ^1	JSR DIVISE	284	BPL <7	
131	BPL <5	208	ADC #\$30	285	BMI <6	
132	STA \$C010	209	STA NUMLIN,Y	286 ^8	JSR \$D72C	
133 ^6	INC INSTR	210	DEY	287	PHP	
134	PLA	211	BPL <1	288	JSR OUTDO	
135	BNE NINSTR	212	INY	289	PLP	
136	JMP BOUCLE	213 ^2	LDA NUMLIN,Y	290	BPL <8	
137 ;		214	CMP #\$30	291	LDA #\$20	
138 ;COMPARE ACC. AU CARACT POINTE		215	BNE >3	292 ^5	JSR OUTDO	
139 COMPAR LDY #\$00		216	LDA #\$20	293	JMP <1	
140	CMP (PTR),Y	217	STA NUMLIN,Y	294 ^9	JSR CR	
141	BNE >1	218	INY	295	RTS	
142	JSR INCPTR1	219	CPY #\$04	296 ;		
143	TYA	220	BNE <2	297 ;TESTE LA FIN D'INSTRUCTION		
144 ^1	RTS	221 ^3	RTS	298 FIN1 JSR INCPTR		
145 ;		222 ;		299	CMP #\$00	
146 ;LIT LE CARACTERE POINTE		223 ;AFFICHAGE D'UN MESSAGE		300	BEQ >1	
147 INCPTR LDY #\$00		224 MESSERR ASL		301	CMP #\$3A	
148	LDA (PTR),Y	225	TAX	302 ^1	RTS	
149 ;		226	INX	303 ;		
150 ;INCREMENTE LE POINTEUR		227	LDA ADRERR,X	304 ;TESTE CARAC. POINTE = LETTRE ?		
151 INCPTR1 INC PTR		228	TAY	305 TESTLET LDY #\$00		
152	BNE >1	229	DEX	306	LDA (PTR),Y	
153	INC PTR+1	230	LDA ADRERR,X	307	CMP #\$41	
154 ^1	RTS	231	JSR STROUT	308	BCC >1	
155 ;		232	RTS	309	CMP #\$5B	
156 ;DECREMENTE LE POINTEUR		233 ;		310	BCS >1	
157 DECPTR SEC		234 ;SORTIE D'UN CARAC. SUR ECRAN,		311	JSR INCPTR1	
158	LDA PTR	235 ;CLIGNOTANT SI CARAC DE CONTROL		312	TYA	
159	SBC #\$01	236 OUTDO ORA #\$80		313	RTS	
160	STA PTR	237	CMP #\$A0	314 ^1	INY	
161	BCS >1	238	BCS >1	315	RTS	
162	DEC PTR+1	239	PHA	316 ;		
163 ^1	RTS	240	STY \$35	317 ;TESTE SI CARAC POINTE = CHIFFR		
164 ;		241	SEC	318 TESTCHI LDY #\$00		
165 ;EMPILE LE POINTEUR		242	SBC #\$40	319	LDA (PTR),Y	
166 EMPILE LDY I		243	JMP \$FDF9	320	CMP #\$30	
167	LDA PTR	244 ^1	JMP COUT1	321	BCC >1	
168	STA PILE,Y	245 ;		322	CMP #\$3A	
169	INY	246 ;LISTE UNE INSTRUCTION BASIC		323	BCS >1	
170	LDA PTR+1	247 LISTING LDY #\$00		324	JSR INCPTR1	
171	STA PILE,Y	248	STY ERR	325	LDY #\$00	
172	INY	249	STY \$85	326	RTS	
173	STY I	250 ^1	LDY \$85	327 ^1	INY	
174	RTS	251	INC \$85	328	RTS	
175 DEPILE DEC I		252	LDA (\$9B),Y	329 ;		
176	DEC I	253	BEQ >9	330 ;TESTE NUMERO DE LIGNE		
177	RTS	254	LDX ERR	331 TNUMLIN LDY #\$00		
178 ;		255	BNE >2	332	STY LNG	
179 ;RESTAURE LE POINTEUR EMPILE		256	CMP #\$3A	333	STY LIN+1	
180 RESTAURE LDY I		257	BEQ >9	334	JSR TESTCHI	

335	BEQ >1	412	LDA LN+1	488	JSR COMPAR
336	LDX #SA1	413	CMP LIN+1	489	BEQ >1
337	RTS	414	BCC <1	490	LDX #SA1
338 ^1	SEC	415	BNE >2	491 ^1	RTS
339	SBC #S30	416	LDA LN	492 TESTAT	LDA #SC5
340	STA LIN	417	CMP LIN	493	JSR COMPAR
341 ^2	JSR TESTCHI	418	BCC <1	494	BEQ >1
342	BEQ >3	419	BNE >2	495	LDX #SC1
343	LDY #S00	420	RTS	496 ^1	RTS
344	RTS	421 ^2	LDX #S03	497 TESTGO	LDA #SAB ;GOTO
345 ^3	SEC	422	RTS	498	JSR COMPAR
346	SBC #S30	423 ;		499	BEQ >1
347	TAY	424 ;TESTE LE TYPE DE VARIABLE		500	LDA #SBO ;GOSUB
348	LDA LNG	425 TESTVAR JSR TESTLET			
349	CLC	426	BNE >3	501	JSR COMPAR
350	ADC #S01	427 ^1	JSR TESTLET	502	BEQ >1
351	CMP #S05	428	BEQ <1	503	LDX #SD1
352	BEQ >4	429	JSR TESTCHI	504 ^1	RTS
353	STA LNG	430	BEQ <1	505 TESTPARO	LDA #S28 ;{
354	LDA LIN	431	LDY #S00	506	JSR COMPAR
355	ASL	432	LDA (PTR),Y	507	BEQ >1
356	STA LIN	433	CMP #S24	508	LDX #S72
357	STA LIN+2	434	BCC >2	509 ^1	RTS
358	LDA LIN+1	435	CMP #S26	510 TESTPARF	LDA #S29 ;}
359	ROL	436	BCS >2	511	JSR COMPAR
360	STA LIN+1	437	TAY	512	BEQ >1
361	ASL LIN+2	438	JSR INCPTR1	513	LDX #S82
362	ROL	439	RTS	514 ^1	RTS
363	ASL LIN+2	440 ^2	LDY #S01	515 ;	
364	ROL	441	RTS	516 ;TESTE SI CHAINE	
365	BCS >4	442 ^3	LDY #S00	517 TESTCHA	LDA #S22 ;"
366	STA LIN+3	443	RTS	518	JSR COMPAR
367	LDA LIN+2	444 ;		519	BNE >3
368	ADC LIN	445 ;TESTE SI VARIABLE REELLE		520 ^1	JSR INCPTR
369	STA LIN	446 TESTVR JSR TESTVAR		521	CMP #S00
370	LDA LIN+3	447	CPY #S00	522	BNE >2
371	ADC LIN+1	448	BNE >1	523	JSR DECPTR
372	BCS >4	449	LDX #S22	524	LDA #S01
373	STA LIN+1	450	RTS	525	RTS
374	TYA	451 ^1	CPY #S01	526 ^2	CMP #S22
375	ADC LIN	452	BEQ >2	527	BNE <1
376	STA LIN	453	LDX #S06	528 ^3	RTS
377	LDA #S00	454 ^2	RTS	529 ;	
378	ADC LIN+1	455 ;		530 ;TESTE UNE CONNECTION	
379	BCS >4	456 ;TESTE SI VARIABLE NUMERIQUE		531 ;DE COMPARAISON	
380	STA LIN+1	457 TESTVNU JSR TESTVAR		532 TESTCOMP	LDA #S00
381	CMP #SFA	458	CPY #S01	533	STA SUP
382	BCC <2	459	BEQ >2	534	STA EGA
383 ^4	LDX #S03	460	CPY #S25	535	STA INF
384	RTS	461	BEQ >2	536 ^1	LDA #SCF ;>
385 ;		462	CPY #S00	537	JSR COMPAR
386 ;TESTE NUMERO LIGNE ET VERIFIE		463	BEQ >1	538	BNE >2
387 ;SON EXISTENCE DANS LE PROGME		464	LDX #S06	539	INC SUP
388 VNUMLIN JSR TNUMLIN		465	RTS	540	JMP <1
389	BNE >2	466 ^1	LDX #S22	541 ^2	LDA #SD0 ;"
390	LDA \$67	467 ^2	RTS	542	JSR COMPAR
391	STA LIN+2	468 ;		543	BNE >3
392	LDA \$68	469 ;SOUS-PROGRAMMES QUI TESTENT		544	INC EGA
393	STA LIN+3	470 ;L'EGALITE DU CARACTERE POINTE		545	JMP <1
394 ^1	LDY #S00	471 ;AVEC DIVERSES CLES		546 ^3	LDA #SD1 ;<
395	LDA (LIN+2),Y	472 TESTVIR LDA #S2C ;,		547	JSR COMPAR
396	STA NL	473	JSR COMPAR	548	BNE >4
397	INY	474	BEQ >1	549	INC INF
398	LDA (LIN+2),Y	475	LDX #S52	550	JMP <1
399	STA NL+1	476 ^1	RTS	551 ^4	LDA #S01
400	ORA NL	477 TESTPOVI LDA #S3B ;;		552	CMP SUP
401	BEQ >2	478	JSR COMPAR	553	BCC >5
402	INY	479	BEQ >1	554	CMP EGA
403	LDA (LIN+2),Y	480	LDX #S91	555	BCC >5
404	STA LN	481 ^1	RTS	556	CMP INF
405	INY	482 TESTEGAL LDA #SD0 ;"		557	BCC >5
406	LDA (LIN+2),Y	483	JSR COMPAR	558	LDA #S00
407	STA LN+1	484	BEQ >1	559	CLC
408	LDA NL	485	LDX #S31	560	ADC SUP
409	STA LIN+2	486 ^1	RTS	561	ADC EGA
410	LDA NL+1	487 TESTTO LDA #SC1 ;TO		562	ADC INF
411	STA LIN+3			563	CMP #S01

564	BEQ >6	641	BEQ >4	718	BEQ SUITE
565	CMP #502	642	LDA #52E	719	JSR RESTAURE
566	BEQ >6	643	JSR COMPAR	720	JSR EXCHPAR
567 ^5	LDA #501	644	BEQ >4	721	BEQ SUITE
568 ^6	RTS	645	TYA	722	RTS
569 ;		646	RTS	723 ;	
570 ;TESTE COMPARAISON DE CHAINE		647 ^4	INY	724 ;CMPARE SGN INT ABS PDL SQR RND	
571 COMPCHA JSR EXPCHA		648	RTS	725 ;LOG EXP COS SIN TAN ATN PEEK	
572	BNE >2	649 ;		726 ^1	CMP #5D2
573	JSR TESTCOMP	650 ;TESTE UNE EXPRESSION NUMERIQUE		727	BCC >1
574	BEQ >1	651 ;ENTRE PARENTHESES		728	CMP #5E3
575	LDX #506	652 EXNUPAR JSR TESTPARO		729	BCS >1
576	RTS	653	BNE >1	730	JSR EXNUPAR
577 ^1	JSR EXPCHA	654 EXNUPARI JSR EXPNUM		731	BEQ SUITE
578 ^2	RTS	655	BNE >1	732	RTS
579 ;		656	JSR TESTPARF	733 ^1	CMP #52E ;.
580 ;TESTE UN OPERATEUR		657 ^1	RTS	734	BNE >1
581 ;+ - * / ^ AND OR		658 ;		735	JSR LITDEC
582 TESTOP LDY #500		659 ;TESTE UNE EXPRESSION NUMERIQUE		736	BEQ SUITE
583	LDA (PTR),Y	660 EXPNUM JSR INCPTR		737	LDX #504
584	CMP #5C8	661	CMP #5C8 ;+	738	RTS
585	BCC >1	662	BEQ EXPNUM	739 ;	
586	CMP #5CF	663	CMP #5C9 ;-	740 ;TESTE CHIFFRE	
587	BCS >1	664	BEQ EXPNUM	741 ^1	CMP #530
588	JSR INCPTR1	665	CMP #5C6 ;NOT	742	BCC >1
589	LDY #500	666	BEQ EXPNUM	743	CMP #53A
590	RTS	667	CMP #528 ;{	744	BCS >1
591 ^1	INY	668	BNE >1	745	JSR LITNOM
592	RTS	669	JSR EXNUPARI	746	BEQ SUITE
593 ;		670	BNE >2	747	LDX #504
594 ;TESTE SI INDICES		671	JMP SUITE	748	RTS
595 ;(POUR VARIABLE TABLEAU)		672 ^2		749 ^1	PHA
596 INDICES LDA #528 ;{		673 ^1	CMP #5C2 ;FN	750	JSR DECPTR
597	JSR COMPAR	674	BNE >1	751	PLA
598	BEQ INDICES1	675	JSR TESTVR	752	CMP #522 ;"
599	LDA #500	676	BNE >2	753	BEQ >4
600	CLC	677	JSR EXNUPAR	754	CMP #5E4 ;STR5
601	RTS	678	BNE >2	755	BEQ >4
602 INDICES1 JSR EXPNUM		679	JMP SUITE	756 ;	
603	BNE >2	680 ^2	RTS	757 ;TESTE CHR\$ LEFT\$ RIGHT\$ MID\$	
604 ^1 LDA #529 ;}		681 ^1	CMP #5D7 ;SCRN(758	CMP #5E7
605	JSR COMPAR	682	BNE >1	759	BCC >1
606	BEQ >2	683	JSR EXPNUM	760	CMP #5EB
607	LDA #52C ;,	684	BNE >2	761	BCC >4
608	JSR COMPAR	685	JSR TESTVIR	762 ^1	JSR EMPILE
609	BNE >3	686	BNE >2	763	JSR TESTVAR
610	JSR EXPNUM	687	JSR EXNUPARI	764	JSR DEPILE
611	BEQ <1	688	BNE >2	765	CPY #524
612 ^2	SEC	689	JMP SUITE	766	BNE >2
613	RTS	690 ^2	RTS	767	JSR RESTAURE
614 ^3	LDX #552	691 ^1	CMP #5E3 ;LEN	768	JMP >4
615	RTS	692	BEQ >2	769 ^2	CPY #500
616 ;		693	CMP #5E5 ;VAL	770	BNE >3
617 ;LIT UN NOMBRE		694	BEQ >2	771	LDX #562
618 ;COMMENCANT PAR UN CHIFFRE		695	CMP #5E6 ;ASC	772	RTS
619 LITNOM JSR TESTCHI		696	BNE >1	773 ^3	JSR INDICES
620	BEQ LITNOM	697 ^2	JSR EXCHPAR	774	BEQ SUITE
621	LDA #52E ;.	698	BNE >3	775	RTS
622	JSR COMPAR	699	JMP SUITE	776 ^4	JSR COMPCHA
623	BNE >1	700 ^3	RTS	777	BEQ SUITE
624 ;		701 ^1	CMP #5D5 ;USR	778	RTS
625 ;LIT UN NOMBRE		702	BNE >1	779 SUITE	JSR TESTCOMP
626 ;COMMENCANT PAR UN POINT		703	JSR EXNUPAR	780	BEQ >1
627 LITDEC JSR TESTCHI		704	BNE >2	781	JSR TESTOP
628	BEQ LITDEC	705	LDX #50B	782	BNE >2
629 ^1	LDA #545 ;E	706	STX ERR	783 ^1	JMP EXPNUM
630	JSR COMPAR	707	JMP SUITE	784 ^2	LDA #500
631	BNE >3	708 ^2	RTS	785	RTS
632	LDA #5C8 ;+	709 ^1	CMP #5D6 ;FRE	786 ;	
633	JSR COMPAR	710	BEQ >2	787 ;TESTE UNE EXPRESSION ALPHANUM.	
634	BEQ >2	711	CMP #5D9 ;POS	788 ;ENTRE PARENTHESES	
635	LDA #5C9 ;-	712	BNE >1	789 EXCHPAR JSR TESTPARO	
636	JSR COMPAR	713 ^2	JSR EMPILE	790	BNE >1
637 ^2	JSR TESTCHI	714	JSR EXNUPAR	791	JSR EXPCHA
638	BNE >4	715	PHP	792	BNE >1
639	JSR TESTCHI	716	JSR DEPILE	793	JSR TESTPARF
640 ^3	JSR TESTCHI	717	PLP	794 ^1	RTS

795 ;		872	STA \$9B	949 ^1	JSR FIN1
796 ;TESTE UNE EXPRESSION ALPHANUM.		873	LDA PTR+1	950	BNE DATA
797 EXPCHA JSR INCPTR		874	STA \$9C	951	JSR DECPTR
798 CMP #\$C8 ;+		875 TRALINS1 JSR INCPTR		952	LDX #\$0D
799 BEQ EXPCHA		876 CMP #\$00 ;FINL		953	JMP FININST
800 CMP #\$E4 ;STR\$		877 BEQ >1		954 ^2	LDX #\$12
801 BEQ >2		878 CMP #\$3A		955	JMP FININST
802 CMP #\$E7 ;CHR\$		879 BNE >2		956 INPUT	LDA #\$22
803 BNE >1		880 ^1 LDX #\$08		957	JSR COMPAR
804 ^2 JSR EXNUPAR		881 RTS		958	BNE >1
805 BNE >3		882 ^2 CMP #\$41 ;A..Z		959	JSR DECPTR
806 JMP SUITEC		883 BCC >3		960	JSR TESTCHA
807 ^3 RTS		884 CMP #\$5B		961	BNE >5
808 ^1 CMP #\$E8 ;LFT\$		885 BCS >3		962	JSR TESTPOVI
809 BEQ >2		886 JSR DECPTR ;LET		963	BNE >4
810 CMP #\$E9 ;RGTS		887 LDA \$8AA		964 ^1	JSR TESTVAR
811 BNE >1		888 ^3 CMP #\$80		965	CPY #\$00
812 ^2 JSR TESTPARO		889 BCC >4		966	BEQ >3
813 BNE >3		890 CMP #\$C0		967	JSR INDICES
814 JSR EXPCHA		891 BCC >5		968	BNE >4
815 BNE >3		892 ^4 JSR FININST		969	JSR TESTV1R
816 JSR TESTVIR		893 LDX #\$00		970	BEQ <1
817 BNE >3		894 RTS		971	JMP END
818 JSR EXNUPAR1		895 ^5 AND #\$7F		972 ^3	LDX #\$22
819 BEQ SUITEC		896 ASL		973 ^4	JMP FININST
820 ^3 RTS		897 TAX		974 ^5	LDX #\$12
821 ^1 CMP #\$EA ;MID\$		898 LDA TABLE,X		975	JMP FININST
822 BNE >1		899 STA IND		976 DEL	JSR TNUMLIN
823 JSR TESTPARO		900 INX		977	BNE >1
824 BNE >2		901 LDA TABLE,X		978	JSR TESTV1R
825 JSR EXPCHA		902 STA IND+1		979	BNE >1
826 BNE >2		903 JMP (IND)		980	JSR TNUMLIN
827 JSR TESTV1R		904 ;		981	BNE >1
828 BNE >2		905 ;TRAITEMENT DIFFERENCIE SELON		982	JMP END
829 JSR EXPNUM		906 ;L'INSTRUCTION BASIC		983 ^1	JMP FININST
830 BNE >2		907 END LDX #\$07		984 DIM	JSR TESTVAR
831 LDA #\$29 ;)		908 END1 JSR FINI		985	CPY #\$00
832 JSR COMPAR		909 BEQ >1		986	BEQ >1
833 BEQ SUITEC		910 JSR FININST		987	JSR INDICES
834 JSR TESTVIR		911 LDX #\$12		988	BNE >2
835 BNE >2		912 ^1 RTS		989	BCC >1
836 JSR EXNUPAR1		913 FOR..TO JSR TESTVR		990	JSR TESTVIR
837 BEQ SUITEC		914 BNE >2		991	BEQ DIM
838 ^2 RTS		915 JSR TESTEGAL		992	JMP END
839 ^1 CMP #\$22		916 BNE >2		993 ^1	LDX #\$B1
840 BNE >1		917 JSR EXPNUM		994 ^2	JMP FININST
841 JSR DECPTR		918 BNE >2		995 READ	JSR TESTVAR
842 JSR TESTCHA		919 JSR TESTTO		996	CPY #\$00
843 BEQ SUITEC		920 BNE >2		997	BEQ >1
844 LDX #\$62		921 JSR EXPNUM		998	JSR INDICES
845 RTS		922 BNE >2		999	BNE >2
846 ^1 CMP #\$41 ;A..Z		923 LDA #\$C7 ;STEP		1000	JSR TESTVIR
847 BCC >1		924 JSR COMPAR		1001	BEQ READ
848 CMP #\$5B		925 BNE >1		1002	JMP END
849 BCS >1		926 JSR EXPNUM		1003 ^1	LDX #\$22
850 JSR DECPTR		927 BNE >2		1004 ^2	JMP FININST
851 JSR TESTVAR		928 ^1 JMP END		1005 CALL	LDX #\$0A
852 CPY #\$24		929 ^2 JMP FININST		1006	JMP FININST
853 BNE >2		930 NEXT JSR TESTLET		1007 PLOT	JSR EXPNUM
854 JSR INDICES		931 BNE >2		1008	BNE >1
855 BEQ SUITEC		932 JSR DECPTR		1009	JSR TESTVIR
856 RTS		933 ^1 JSR TESTVR		1010	BNE >1
857 ^2 LDX #\$06		934 BNE >3		1011	JSR EXPNUM
858 RTS		935 JSR TESTVIR		1012	BNE >1
859 ^1 JSR DECPTR		936 BEQ <1		1013	JMP END
860 JSR EXCHPAR		937 ^2 JMP END		1014 ^1	JMP FININST
861 BEQ SUITEC		938 ^3 JMP FININST		1015 HLIN	JSR EXPNUM
862 RTS		939 DATA LDA \$22		1016	BNE >1
863 SUITEC LDA #\$C8 ;+		940 JSR COMPAR		1017	JSR TESTV1R
864 JSR COMPAR		941 BNE >1		1018	BNE >1
865 BNE >1		942 JSR DECPTR		1019	JSR EXPNUM
866 JMP EXPCHA		943 JSR TESTCHA		1020	BNE >1
867 ^1 LDA #\$00		944 BNE >2		1021	JSR TESTAT
868 RTS		945 JSR TESTVIR		1022	BNE >1
869 ;		946 BEQ DATA		1023	JSR EXPNUM
870 ;TRAITEMENT D'UNE INSTRUCTION		947 LDX #\$0D		1024	BNE >1
871 TRALINS LDA PTR		948 JMP END1		1025	JMP END

1026 ^1	JMP FININST	1103	JSR VNUMLIN	1180	PLP
1027 HCOLOR	JSR EXPNUM	1104	BEQ >4	1181	BEQ PRINT
1028	BNE >1	1105	BNE >5	1182	JSR RESTAURE
1029	JMP END	1106 ^1	LDA #SC4	1183	JSR EMPILE
1030 ^1	JMP FININST	1107	JSR COMPAR	1184	JSR EXPCHA
1031 HPILOT	JSR TESTTO	1108	BEQ >2	1185	PHP
1032 ^1	JSR EXPNUM	1109	LDX #SE1	1186	JSR DEPILE
1033	BNE >2	1110	BNE >5	1187	PLP
1034	JSR TESTVIR	1111 ^2	JSR TESTCHI	1188	BEQ PRINT
1035	BNE >2	1112	BEQ >3	1189	JSR RESTAURE
1036	JSR EXPNUM	1113	JMP TRAIINS1	1190	JMP END
1037	BNE >2	1114 ^3	JSR DECPTR	1191 ^3	JMP FININST
1038	JSR TESTTO	1115	JSR VNUMLIN	1192 LIST	JSR TESTCHI
1039	BEQ <1	1116	BNE >5	1193	BNE >1
1040	JMP END	1117 ^4	JMP END	1194	JSR DECPTR
1041 ^2	JMP FININST	1118 ^5	JMP FININST	1195	JSR TNUMLIN
1042 DRAW	JSR EXPNUM	1119 AMPERS	LDX #S05	1196	BNE >4
1043	BNE >2	1120	JMP FININST	1197 ^1	JSR TESTVIR
1044	JSR TESTAT	1121 REM	LDX #S0C	1198	BEQ >2
1045	BNE >1	1122	JMP FINBIS	1199	LDA #SC9
1046	JSR EXPNUM	1123 ON	JSR EXPNUM	1200	JSR COMPAR
1047	BNE >2	1124	BNE >2	1201	BNE >3
1048	JSR TESTVIR	1125	JSR TESTGO	1202 ^2	JSR TESTCHI
1049	BNE >2	1126	BNE >2	1203	BNE >3
1050	JSR EXPNUM	1127 ^1	JSR VNUMLIN	1204	JSR DECPTR
1051	BNE >2	1128	BNE >2	1205	JSR TNUMLIN
1052 ^1	JMP END	1129	JSR TESTVIR	1206	BNE >4
1053 ^2	JMP FININST	1130	BEQ <1	1207 ^3	JMP END
1054 ONERR	LDA #SAB	1131	JMP END	1208 ^4	JMP FININST
1055	JSR COMPAR	1132 ^2	JMP FININST	1209 GET	LDY #S07
1056	BNE >1	1133 WAIT	JSR EXPNUM	1210	STY DRGET
1057	JSR VNUMLIN	1134	BNE >2	1211 ^1	JSR TESTVAR
1058	BNE >2	1135	JSR TESTVIR	1212	CPY #S00
1059	JMP END	1136	BNE >2	1213	BNE >2
1060 ^1	LDX #SD1	1137	JSR EXPNUM	1214	LDX #S22
1061 ^2	JMP FININST	1138	BNE >2	1215	BNE >4
1062 RECALL	JSR TESTVNU	1139	JSR TESTVIR	1216 ^2	CPY #S24
1063	BNE >1	1140	BNE >1	1217	BEQ >3
1064	JMP END	1141	JSR EXPNUM	1218	LDA #S0E
1065 ^1	JMP FININST	1142	BNE >2	1219	STA DRGET
1066 LET	JSR TESTVAR	1143 ^1	JMP END	1220 ^3	JSR INDICES
1067	CPY #S00	1144 ^2	JMP FININST	1221	BNE >4
1068	BEQ >2	1145 DEF	LDA #SC2	1222	JSR TESTVIR
1069	CPY #S24	1146	JSR COMPAR	1223	BEQ <1
1070	BEQ >1	1147	BNE >1	1224	LDX DRGET
1071	JSR INDICES	1148	JSR TESTVR	1225	JMP END1
1072	BNE >3	1149	BNE >2	1226 ^4	JMP FININST
1073	JSR TESTEGAL	1150	JSR TESTPARO	1227 ;	
1074	BNE >3	1151	BNE >2	1228 ;PLACE LE POINTEUR AU DEBUT	
1075	JSR EXPNUM	1152	JSR TESTVR	1229 ;INSTRUCTION SUIVANTE, APRES :	
1076	BNE >3	1153	BNE >2	1230 ;OU NOUVELLE LIGNE	
1077	JMP END	1154	JSR TESTPARF	1231 FININST JSR INCPTR	
1078 ^1	JSR INDICES	1155	BNE >2	1232	CMP #S00
1079	BNE >3	1156	JSR TESTEGAL	1233	BEQ >1
1080	JSR TESTEGAL	1157	BNE >2	1234	CMP #S3A
1081	BNE >3	1158	JSR EXPNUM	1235	BNE FININST
1082	JSR EXPCHA	1159	BNE >2	1236 ^1	RTS
1083	BNE >3	1160	JMP END	1237 FINBIS JSR INCPTR	
1084	JMP END	1161 ^1	LDX #SF1	1238	CMP #S00
1085 ^2	LDX #S22	1162 ^2	JMP FININST	1239	BNE FINBIS
1086 ^3	JMP FININST	1163 PRINT	JSR TESTVIR	1240	RTS
1087 GOTO	JSR VNUMLIN	1164	BEQ PRINT	1241 ;	
1088	BNE >1	1165	JSR TESTPOV1	1242 ;DONNEES : MESSAGES	
1089	JMP END	1166	BEQ PRINT	1243 TXLIGNE ASC "LIGNE "	
1090 ^1	JMP FININST	1167	LDA #SCO	1244 NUMLIN HEX 000000000000	
1091 RUN	JSR TESTCHI	1168	JSR COMPAR	1245 ERRO ASC "INSTRUCTION NON	
1092	BNE >1	1169	BEQ >1	RECONNUE"	
1093	JSR DECPTR	1170	LDA #SC3	1246	HEX 0D00
1094	JSR VNUMLIN	1171	JSR COMPAR	1247 ERR1 ASC " ATTENDU"	
1095	BEQ >1	1172	BNE >2	1248	HEX 0D00
1096	JMP FININST	1173 ^1	JSR EXNUPAR1	1249 ERR2 ASC " ATTENDUE"	
1097 ^1	JMP END	1174	BNE >3	1250	HEX 0D00
1098 IF	JSR EXPNUM	1175	BEQ PRINT	1251 ERR3 ASC "MAUVAISE ADRESSE"	
1099	BNE >5	1176 ^2	JSR EMPILE	1252	HEX 0D00
1100	LDA #SAB	1177	JSR EXPNUM	1253 ERR4 ASC "NOMBRE INCORRECT"	
1101	JSR COMPAR	1178	PHP	1254	HEX 0D00
1102	BNE >1	1179	JSR DEPILE	1255 ERR5 ASC "EXPRESSION	

```

INCORRECTE"
1256      HEX 0D00
1257 ERR6  ASC "CONFUSION DE
TYPE"
1258      HEX 0D00
1259 ERR7  ASC "CORRECT"
1260      HEX 0D00
1261 ERR8  ASC "VIDE"
1262      HEX 0D00
1263 ERR9  ASC "INSTRUCTION &"
1264      HEX 0D00
1265 ERR10  ASC "INSTRUCTION CALL"
1266      HEX 0D00
1267 ERR11  ASC "FONCTION USR"
1268      HEX 0D00
1269 ERR12  ASC "INSTRUCTION REM"
1270      HEX 0D00
1271 ERR13  ASC "INSTRUCTION DATA"
1272      HEX 0D00
1273 ERR14  ASC "GET NUMERIQUE"
1274      HEX 0D00
1275 ERR17  ASC "FIN D'INSTR."
1276      HEX 00
1277 ERR18  ASC "VARIABLE"
1278      HEX 00
1279 ERR19  ASC "="
1280      HEX 00
1281 ERR20  ASC "TO"
1282      HEX 00
1283 ERR21  ASC ", "
1284      HEX 00
1285 ERR22  ASC "EXPRESSION"
1286      HEX 00
1287 ERR23  ASC "{"
1288      HEX 00
1289 ERR24  ASC "}"
1290      HEX 00
1291 ERR25  ASC ";"
1292      HEX 00
1293 ERR26  ASC "NUMERO"
1294      HEX 00
1295 ERR27  ASC "TABLEAU"
1296      HEX 00
1297 ERR28  ASC "AT"
1298      HEX 00
1299 ERR29  ASC "GOTO OU GOSUB"
1300      HEX 00
1301 ERR30  ASC "THEN"
1302      HEX 00
1303 ERR31  ASC "FN"
1304      HEX 00
1305 ;
1306 ;TABLE DES ADRESSES DES DEBUTS
1307 ;DE MESSAGES
1308 ADRERR  ADR ERRO
1309      ADR ERR1
1310      ADR ERR2
1311      ADR ERR3
1312      ADR ERR4
1313      ADR ERR5
1314      ADR ERR6
1315      ADR ERR7
1316      ADR ERR8
1317      ADR ERR9
1318      ADR ERR10
1319      ADR ERR11
1320      ADR ERR12
1321      ADR ERR13
1322      ADR ERR14
1323      ADR ERR17
1324      ADR ERR18
1325      ADR ERR19
1326      ADR ERR20
1327      ADR ERR21
1328      ADR ERR22
1329      ADR ERR23
1330      ADR ERR24

```

```

1331      ADR ERR25
1332      ADR ERR26
1333      ADR ERR27
1334      ADR ERR28
1335      ADR ERR29
1336      ADR ERR30
1337      ADR ERR31
1338 ;
1339 ;TABLE DES ADRESSES DES
1340 ;TRAITEMENTS SPECIQUES
1341 TABLE  ADR END
1342      ADR FOR..TO
1343      ADR NEXT
1344      ADR DATA
1345      ADR INPUT
1346      ADR DEL
1347      ADR DIM
1348      ADR READ
1349      ADR END ;GR
1350      ADR END ;TEXT
1351      ADR HCOLOR ;PR#
1352      ADR HCOLOR ;LN#
1353      ADR CALL
1354      ADR PLOT
1355      ADR HLIN
1356      ADR HLIN ;VLIN
1357      ADR END ;HGR2
1358      ADR END ;HGR
1359      ADR HCOLOR
1360      ADR HPLOT
1361      ADR DRAW
1362      ADR DRAW ;XDRW
1363      ADR HCOLOR ;HTAB
1364      ADR END ;HOME
1365      ADR HCOLOR ;ROT=
1366      ADR HCOLOR ;SCL=
1367      ADR END ;SHLD
1368      ADR END ;TRAC
1369      ADR END ;NOTR
1370      ADR END ;NORM
1371      ADR END ;INVE
1372      ADR END ;FLSH
1373      ADR HCOLOR ;CLR=
1374      ADR END ;POP
1375      ADR HCOLOR ;VTAB
1376      ADR HCOLOR ;HIM:
1377      ADR HCOLOR ;LOM:
1378      ADR ONERR
1379      ADR END ;RESU
1380      ADR RECALL
1381      ADR RECALL ;STOR
1382      ADR HCOLOR ;SPED
1383      ADR LET
1384      ADR GOTO
1385      ADR RUN
1386      ADR IF
1387      ADR END ;REST
1388      ADR AMPERS
1389      ADR GOTO ;GOSU
1390      ADR END ;RTRN
1391      ADR REM
1392      ADR END ;STOP
1393      ADR ON
1394      ADR WAIT
1395      ADR END ;LOAD
1396      ADR END ;SAVE
1397      ADR DEF
1398      ADR PLOT ;POKE
1399      ADR PRINT
1400      ADR END ;CONT
1401      ADR LIST
1402      ADR END ;CLEA
1403      ADR GET
1404      ADR END ;NEW
1405 IND  HEX 0000
1406      END

```

Récapitulation SNTX

```

8900- 20 58 FC A9 4C 8D F5 03
8908- A9 00 85 73 8D F6 03 A9
8910- 89 85 74 8D F7 03 A5 67
8918- 85 06 A5 68 85 07 20 CE
8920- 89 85 08 20 CE 89 85 09
8928- 05 08 D0 03 4C D0 03 20
8930- CE 89 85 18 20 CE 89 85
8938- 19 20 22 8A 20 80 FE A9
8940- 6A A0 92 20 3A DB A9 01
8948- 85 CE A9 0B 85 24 20 80
8950- FE A5 CE 85 18 A9 00 85
8958- 19 20 22 8A A9 73 A0 92
8960- 20 3A DB 20 84 FE E6 24
8968- A2 00 86 1D 86 CF 20 E5
8970- 8E 48 E0 07 D0 06 A4 CF
8978- F0 02 A2 0B 8A 29 0F 48
8980- C9 01 F0 04 C9 02 D0 0C
8988- 48 8A 4A 4A 4A 4A 69 0E
8990- 20 42 8A 68 20 42 8A 68
8998- 48 C9 07 F0 0B C9 08 F0
89A0- 07 C9 0C F0 03 20 63 8A
89A8- 68 C9 07 B0 0E 20 3A FF
89B0- 8D 10 C0 AD 00 C0 10 FB
89B8- 8D 10 C0 E6 CE 68 D0 8A
89C0- 4C 1E 89 A0 00 D1 06 D0
89C8- 04 20 D2 89 98 60 A0 00
89D0- B1 06 E6 06 D0 02 E6 07
89D8- 60 38 A5 06 E9 01 85 06
89E0- B0 02 C6 07 60 A4 1D A5
89E8- 06 99 00 03 C8 A5 07 99
89F0- 00 03 C8 84 1D 60 C6 1D
89F8- C6 1D 60 A4 1D B9 00 03
8A00- 85 06 C8 B9 00 03 85 07
8A08- 60 A2 10 A9 00 06 18 26
8A10- 19 2A C9 0A 90 02 E9 0A
8A18- 26 18 26 19 2A CA D0 F2
8A20- 6A 60 A0 04 20 09 8A 69
8A28- 30 99 70 92 88 10 F5 C8
8A30- B9 70 92 C9 30 D0 0A A9
8A38- 20 99 70 92 C8 C0 04 D0
8A40- EF 60 0A AA E8 BD B8 93
8A48- A8 CA BD B8 93 20 3A DB
8A50- 60 09 80 C9 A0 B0 09 48
8A58- 84 35 38 E9 40 4C F9 FD
8A60- 4C F0 FD A0 00 84 CF 84
8A68- 85 A4 85 E6 85 B1 9B F0
8A70- 4F A6 CF D0 04 C9 3A F0
8A78- 47 C9 22 D0 08 48 A5 CF
8A80- 49 01 85 CF 68 C9 B2 D0
8A88- 04 A2 FF 86 CF C9 80 90
8A90- 29 38 E9 7F AA A9 20 20
8A98- 51 8A A0 D0 84 9D A0 CF
8AA0- 84 9E A0 FF CA F0 07 20
8AA8- 2C D7 10 FB 30 F6 20 2C
8AB0- D7 08 20 51 8A 28 10 F6
8AB8- A9 20 20 51 8A 4C 69 8A
8AC0- 20 62 FC 60 20 CE 89 C9
8AC8- 00 F0 02 C9 3A 60 A0 00

```

8AD0-	B1 06 C9 41 90 09 C9 5B	8CB8-	C9 C8 90 0A C9 CF B0 06	8EA0-	31 8D F0 34 60 C9 22 D0
8AD8-	B0 05 20 D2 89 98 60 C8	8CC0-	20 D2 89 A0 00 60 C8 60	8EA8-	0B 20 D9 89 20 3C 8C F0
8AE0-	60 A0 00 B1 06 C9 30 90	8CC8-	A9 28 20 C3 89 F0 04 A9	8EB0-	27 A2 62 60 C9 41 90 17
8AE8-	0A C9 3A B0 06 20 D2 89	8CD0-	00 18 60 20 3A 8D D0 13	8EB8-	C9 5B B0 13 20 D9 89 20
8AF0-	A0 00 60 C8 60 A0 00 84	8CD8-	A9 29 20 C3 89 F0 0C A9	8EC0-	99 8B C0 24 D0 06 20 C8
8AF8-	EE 84 FB 20 E1 8A F0 03	8CE0-	2C 20 C3 89 D0 07 20 3A	8EC8-	8C F0 0D 60 A2 06 60 20
8B00-	A2 A1 60 38 E9 30 85 FA	8CE8-	8D F0 ED 38 60 A2 52 60	8ED0-	D9 89 20 38 8E F0 01 60
8B08-	20 E1 8A F0 03 A0 00 60	8CF0-	20 E1 8A F0 FB A9 2E 20	8ED8-	A9 C8 20 C3 89 D0 03 4C
8B10-	38 E9 30 A8 A5 EE 18 69	8CF8-	C3 89 D0 05 20 E1 8A F0	8EE0-	46 8E A9 00 60 A5 06 85
8B18-	01 C9 05 F0 37 85 EE A5	8D00-	FB A9 45 20 C3 89 D0 14	8EE8-	9B A5 07 85 9C 20 CE 89
8B20-	FA 0A 85 FA 85 FC A5 FB	8D08-	A9 C8 20 C3 89 F0 05 A9	8EF0-	C9 00 F0 04 C9 3A D0 03
8B28-	2A 85 FB 06 FC 2A 06 FC	8D10-	C9 20 C3 89 20 E1 8A D0	8EF8-	A2 08 60 C9 41 90 09 C9
8B30-	2A B0 21 85 FD A5 FC 65	8D18-	11 20 E1 8A 20 E1 8A F0	8F00-	5B B0 05 20 D9 89 A9 AA
8B38-	FA 85 FA A5 FD 65 FB B0	8D20-	09 A9 2E 20 C3 89 F0 02	8F08-	C9 80 90 04 C9 C0 90 06
8B40-	13 85 FB 98 65 FA 85 FA	8D28-	98 60 C8 60 20 28 8C D0	8F10-	20 56 92 A2 00 60 29 7F
8B48-	A9 00 65 FB B0 06 85 FB	8D30-	08 20 3A 8D D0 03 20 32	8F18-	0A AA BD F4 93 8D 74 94
8B50-	C9 FA 90 B4 A2 03 60 20	8D38-	8C 60 20 CE 89 C9 C8 F0	8F20-	E8 BD F4 93 8D 75 94 6C
8B58-	F5 8A D0 3A A5 67 85 FC	8D40-	F9 C9 C9 F0 F5 C9 C6 F0	8F28-	74 94 A2 07 20 C4 8A F0
8B60-	A5 68 85 FD A0 00 B1 FC	8D48-	F1 C9 28 D0 09 20 31 8D	8F30-	05 20 56 92 A2 12 60 20
8B68-	85 08 C8 B1 FC 85 09 05	8D50-	D0 03 4C 28 8E 60 C9 C2	8F38-	BF 8B D0 23 20 F9 8B D0
8B70-	08 F0 23 C8 B1 FC 85 18	8D58-	D0 0E 20 BF 8B D0 08 20	8F40-	1E 20 3A 8D D0 19 20 03
8B78-	C8 B1 FC 85 19 A5 08 85	8D60-	2C 8D D0 03 4C 28 8E 60	8F48-	8C D0 14 20 3A 8D D0 0F
8B80-	FC A5 09 85 FD A5 19 C5	8D68-	C9 D7 D0 13 20 3A 8D D0	8F50-	A9 C7 20 C3 89 D0 05 20
8B88-	FB 90 D9 D0 09 A5 18 C5	8D70-	0D 20 E5 8B D0 08 20 31	8F58-	3A 8D D0 03 4C 2A 8F 4C
8B90-	FA 90 D1 D0 01 60 A2 03	8D78-	8D D0 03 4C 28 8E 60 C9	8F60-	56 92 20 CE 8A D0 0D 20
8B98-	60 20 CE 8A D0 1E 20 CE	8D80-	E3 F0 08 C9 E5 F0 04 C9	8F68-	D9 89 20 BF 8B D0 08 20
8BA0-	8A F0 FB 20 E1 8A F0 F6	8D88-	E6 D0 09 20 38 8E D0 03	8F70-	E5 8B F0 F6 4C 2A 8F 4C
8BA8-	A0 00 B1 06 C9 24 90 09	8D90-	4C 28 8E 60 C9 D5 D0 0D	8F78-	56 92 A9 22 20 C3 89 D0
8BB0-	C9 26 B0 05 A8 20 D2 89	8D98-	20 2C 8D D0 07 A2 0B 86	8F80-	19 20 D9 89 20 3C 8C D0
8BB8-	60 A0 01 60 A0 00 60 20	8DA0-	CF 4C 28 8E 60 C9 D6 F0	8F88-	1E A9 20 20 C3 89 F0 F9
8BC0-	99 8B C0 00 D0 03 A2 22	8DA8-	04 C9 D9 D0 16 20 E5 89	8F90-	20 E5 8B F0 E5 A2 0D 4C
8BC8-	60 C0 01 F0 02 A2 06 60	8DB0-	20 2C 8D 08 20 F6 89 28	8F98-	2C 8F 20 C4 8A D0 DB 20
8BD0-	20 99 8B C0 01 F0 0D C0	8DB8-	F0 6E 20 FB 89 20 38 8E	8FA0-	D9 89 A2 0D 4C 56 92 A2
8BD8-	25 F0 09 C0 00 F0 03 A2	8DC0-	F0 66 60 C9 D2 90 0A C9	8FA8-	12 4C 56 92 A9 22 20 C3
8BE0-	06 60 A2 22 60 A9 2C 20	8DC8-	E3 B0 06 20 2C 8D F0 58	8FB0-	89 D0 0D 20 D9 89 20 3C
8BE8-	C3 89 F0 02 A2 52 60 A9	8DD0-	60 C9 2E D0 08 20 FC 8C	8FB8-	8C D0 1E 20 EF 8B D0 16
8BF0-	3B 20 C3 89 F0 02 A2 91	8DD8-	F0 4E A2 04 60 C9 30 90	8FC0-	20 99 8B C0 00 F0 0D 20
8BF8-	60 A9 D0 20 C3 89 F0 02	8DE0-	0C C9 3A B0 08 20 F0 8C	8FC8-	C8 8C D0 0A 20 E5 8B F0
8C00-	A2 31 60 A9 C1 20 C3 89	8DE8-	F0 3E A2 04 60 48 20 D9	8FD0-	EF 4C 2A 8F A2 22 4C 56
8C08-	F0 02 A2 41 60 A9 C5 20	8DF0-	89 68 C9 22 F0 2C C9 E4	8FD8-	92 A2 12 4C 56 92 20 F5
8C10-	C3 89 F0 02 A2 C1 60 A9	8DF8-	F0 28 C9 E7 90 04 C9 EB	8FE0-	8A D0 0D 20 E5 8B D0 08
8C18-	AB 20 C3 89 F0 09 A9 B0	8E00-	90 20 20 E5 89 20 99 8B	8FE8-	20 F5 8A D0 03 4C 2A 8F
8C20-	20 C3 89 F0 02 A2 D1 60	8E08-	20 F6 89 C0 24 D0 06 20	8FF0-	4C 56 92 20 99 8B C0 00
8C28-	A9 28 20 C3 89 F0 02 A2	8E10-	FB 89 4C 22 8E C0 00 D0	8FF8-	F0 0F 20 C8 8C D0 0C 90
8C30-	72 60 A9 29 20 C3 89 F0	8E18-	03 A2 62 60 20 C8 8C F0	9000-	08 20 E5 8B F0 ED 4C 2A
8C38-	02 A2 82 60 A9 22 20 C3	8E20-	07 60 20 A3 8C F0 01 60	9008-	8F A2 B1 4C 56 92 20 99
8C40-	89 D0 11 20 CE 89 C9 00	8E28-	20 55 8C F0 05 20 B4 8C	9010-	8B C0 00 F0 0D 20 C8 8C
8C48-	D0 06 20 D9 89 A9 01 60	8E30-	D0 03 4C 3A 8D A9 00 60	9018-	D0 0A 20 E5 8B F0 EF 4C
8C50-	C9 22 D0 EF 60 A9 00 85	8E38-	20 28 8C D0 08 20 46 8E	9020-	2A 8F A2 22 4C 56 92 A2
8C58-	EB 85 EC 85 ED A9 CF 20	8E40-	D0 03 20 32 8C 60 20 CE	9028-	0A 4C 56 92 20 3A 8D D0
8C60-	C3 89 D0 05 E6 EB 4C 5D	8E48-	89 C9 C8 F0 F9 C9 E4 F0	9030-	0D 20 E5 8B D0 08 20 3A
8C68-	8C A9 D0 20 C3 89 D0 05	8E50-	04 C9 E7 D0 09 20 2C 8D	9038-	8D D0 03 4C 2A 8F 4C 56
8C70-	E6 EC 4C 5D 8C A9 D1 20	8E58-	D0 03 4C D8 8E 60 C9 E8	9040-	92 20 3A 8D D0 17 20 E5
8C78-	C3 89 D0 05 E6 ED 4C 5D	8E60-	F0 04 C9 E9 D0 15 20 28	9048-	8B D0 12 20 3A 8D D0 0D
8C80-	8C A9 01 C5 EB 90 19 C5	8E68-	8C D0 0F 20 46 8E D0 0A	9050-	20 0D 8C D0 08 20 3A 8D
8C88-	EC 90 15 C5 ED 90 11 A9	8E70-	20 E5 8B D0 05 20 31 8D	9058-	D0 03 4C 2A 8F 4C 56 92
8C90-	00 18 65 EB 65 EC 65 ED	8E78-	F0 5E 60 C9 EA D0 26 20	9060-	20 3A 8D D0 03 4C 2A 8F
8C98-	C9 01 F0 06 C9 02 F0 02	8E80-	28 8C D0 20 20 46 8E D0	9068-	4C 56 92 20 03 8C 20 3A
8CA0-	A9 01 60 20 46 8E D0 0B	8E88-	1B 20 E5 8B D0 16 20 3A	9070-	8D D0 12 20 E5 8B D0 0D
8CA8-	20 55 8C F0 03 A2 06 60	8E90-	8D D0 11 A9 29 20 C3 89	9078-	20 3A 8D D0 08 20 03 8C
8CB0-	20 46 8E 60 A0 00 B1 06	8E98-	F0 3E 20 E5 8B D0 05 20	9080-	F0 EC 4C 2A 8F 4C 56 92

9088- 20 3A 8D D0 17 20 0D 8C
 9090- D0 0F 20 3A 8D D0 0D 20
 9098- E5 8B D0 08 20 3A 8D D0
 90A0- 03 4C 2A 8F 4C 56 92 A9
 90A8- AB 20 C3 89 D0 08 20 57
 90B0- 8B D0 05 4C 2A 8F A2 D1
 90B8- 4C 56 92 20 D0 8B D0 03
 90C0- 4C 2A 8F 4C 56 92 20 99
 90C8- 8B C0 00 F0 28 C0 24 F0
 90D0- 12 20 C8 8C D0 21 20 F9
 90D8- 8B D0 1C 20 3A 8D D0 17
 90E0- 4C 2A 8F 20 C8 8C D0 0F
 90E8- 20 F9 8B D0 0A 20 46 8E
 90F0- D0 05 4C 2A 8F A2 22 4C
 90F8- 56 92 20 57 8B D0 03 4C
 9100- 2A 8F 4C 56 92 20 E1 8A
 9108- D0 0B 20 D9 89 20 57 8B
 9110- F0 03 4C 56 92 4C 2A 8F
 9118- 20 3A 8D D0 2C A9 AB 20
 9120- C3 89 D0 07 20 57 8B F0
 9128- 1D D0 1E A9 C4 20 C3 89
 9130- F0 04 A2 E1 D0 13 20 E1
 9138- 8A F0 03 4C ED 8E 20 D9
 9140- 89 20 57 8B D0 03 4C 2A
 9148- 8F 4C 56 92 A2 09 4C 56
 9150- 92 A2 0C 4C 62 92 20 3A
 9158- 8D D0 12 20 17 8C D0 0D
 9160- 20 57 8B D0 08 20 E5 8B
 9168- F0 F6 4C 2A 8F 4C 56 92
 9170- 20 3A 8D D0 17 20 E5 8B
 9178- D0 12 20 3A 8D D0 0D 20
 9180- E5 8B D0 05 20 3A 8D D0
 9188- 03 4C 2A 8F 4C 56 92 A9
 9190- C2 20 C3 89 D0 21 20 BF
 9198- 8B D0 1E 20 28 8C D0 19
 91A0- 20 BF 8B D0 14 20 32 8C
 91A8- D0 0F 20 F9 8B D0 0A 20
 91B0- 3A 8D D0 05 4C 2A 8F A2
 91B8- F1 4C 56 92 20 E5 8B F0
 91C0- FB 20 EF 8B F0 F6 A9 C0
 91C8- 20 C3 89 F0 07 A9 C3 20
 91D0- C3 89 D0 07 20 31 8D D0
 91D8- 25 F0 E1 20 E5 89 20 3A
 91E0- 8D 08 20 F6 89 28 F0 D4
 91E8- 20 FB 89 20 E5 89 20 46
 91F0- 8E 08 20 F6 89 28 F0 C4
 91F8- 20 FB 89 4C 2A 8F 4C 56
 9200- 92 20 E1 8A D0 08 20 D9
 9208- 89 20 F5 8A D0 1C 20 E5
 9210- 8B F0 07 A9 C9 20 C3 89
 9218- D0 0D 20 E1 8A D0 08 20
 9220- D9 89 20 F5 8A D0 03 4C
 9228- 2A 8F 4C 56 92 A0 07 84
 9230- 1E 20 99 8B C0 00 D0 04
 9238- A2 22 D0 17 C0 24 F0 04
 9240- A9 0E 85 1E 20 C8 8C D0
 9248- 0A 20 E5 8B F0 E3 A6 1E
 9250- 4C 2C 8F 4C 56 92 20 CE
 9258- 89 C9 00 F0 04 C9 3A D0
 9260- F5 60 20 CE 89 C9 00 D0
 9268- F9 60 CC C9 C7 CE C5 A0

9270- 00 00 00 00 00 00 C9 CE
 9278- D3 D4 D2 D5 C3 D4 C9 CF
 9280- CE A0 CE CF CE A0 D2 C5
 9288- C3 CF CE CE D5 C5 0D 00
 9290- A0 C1 D4 D4 C5 CE C4 D5
 9298- 0D 00 A0 C1 D4 D4 C5 CE
 92A0- C4 D5 C5 0D 00 CD C1 D5
 92A8- D6 C1 C9 D3 C5 A0 C1 C4
 92B0- D2 C5 D3 D3 C5 0D 00 CE
 92B8- CF CD C2 D2 C5 A0 C9 CE
 92C0- C3 CF D2 D2 C5 C3 D4 0D
 92C8- 00 C5 D8 D0 D2 C5 D3 D3
 92D0- C9 CF CE A0 C9 CE C3 CF
 92D8- D2 D2 C5 C3 D4 C5 0D 00
 92E0- C3 CF CE C6 D5 D3 C9 CF
 92E8- CE A0 C4 C5 A0 D4 D9 D0
 92F0- C5 0D 00 C3 CF D2 D2 C5
 92F8- C3 D4 0D 00 D6 C9 C4 C5
 9300- 0D 00 C9 CE D3 D4 D2 D5
 9308- C3 D4 C9 CF CE A0 A6 0D
 9310- 00 C9 CE D3 D4 D2 D5 C3
 9318- D4 C9 CF CE A0 C3 C1 CC
 9320- CC 0D 00 C6 CF CE C3 D4
 9328- C9 CF CE A0 D5 D3 D2 0D
 9330- 00 C9 CE D3 D4 D2 D5 C3
 9338- D4 C9 CF CE A0 D2 C5 CD
 9340- 0D 00 C9 CE D4 D2 D5 C3
 9348- D4 C9 CF CE A0 C4 C1 D4
 9350- C1 0D 00 C7 C5 D4 A0 CE
 9358- D5 CD C5 D2 C9 D1 D5 C5
 9360- 0D 00 C6 C9 CE A0 C4 A7
 9368- C9 CE D3 D4 D2 AE 00 D6
 9370- C1 D2 C9 C1 C2 CC C5 00
 9378- BD 00 D4 CF 00 AC 00 C5

9380- D8 D0 D2 C5 D3 D3 C9 CF
 9388- CE 00 A8 00 A9 00 BB 00
 9390- CE D5 CD C5 D2 CF 00 D4
 9398- C1 C2 CC C5 C1 D5 00 C1
 93A0- D4 00 C7 CF D4 CF A0 CF
 93A8- D5 A0 C7 CF D3 D5 C2 00
 93B0- D4 C8 C5 CE 00 C6 CE 00
 93B8- 76 92 90 92 9A 92 A5 92
 93C0- B7 92 C9 92 E0 92 F3 92
 93C8- FC 92 02 93 11 93 23 93
 93D0- 31 93 42 93 53 93 62 93
 93D8- 6F 93 78 93 7A 93 7D 93
 93E0- 7F 93 8A 93 8C 93 8E 93
 93E8- 90 93 97 93 9F 93 A2 93
 93F0- B0 93 B5 93 2A 8F 37 8F
 93F8- 62 8F 7A 8F AC 8F DE 8F
 9400- F3 8F 0E 90 2A 8F 2A 8F
 9408- 60 90 60 90 27 90 2C 90
 9410- 41 90 41 90 2A 8F 2A 8F
 9418- 60 90 6B 90 88 90 88 90
 9420- 60 90 2A 8F 60 90 60 90
 9428- 2A 8F 2A 8F 2A 8F 2A 8F
 9430- 2A 8F 2A 8F 60 90 2A 8F
 9438- 60 90 60 90 60 90 A7 90
 9440- 2A 8F BB 90 BB 90 60 90
 9448- C6 90 FA 90 05 91 18 91
 9450- 2A 8F 4C 91 FA 90 2A 8F
 9458- 51 91 2A 8F 56 91 70 91
 9460- 2A 8F 2A 8F 8F 91 2C 90
 9468- BC 91 2A 8F 01 92 2A 8F
 9470- 2D 92 2A 8F 00 00 E4 8B
 9478- 31 E2 8B 32 CF 8B 31 C9
 9480- 54

Editeur Plein Ecran EPE

Le Pacha

Apple][+, //e, //c

- Listez vos programmes Basic en avant et en arrière.
- Modifiez, insérez, effacez des caractères en plein écran sans relire les lignes.
- Recherchez toute chaîne de caractères.
- Choisissez vous-même les codes de contrôle d'EPE.
- Modifiez EPE : le fichier source est sur la disquette.

150,00 F TTC franco
(bon de commande page 74)

Un catalogue Multi-SED

Ce programme, rédigé entièrement en Applesoft sous DOS 3.3, permet d'afficher le contenu de toute disquette formatée sous DOS 3.3. ProDOS, p-UCSD ou CP/M 80, ceci sans nécessairement savoir au départ quel est son système d'exploitation. Si la disquette n'est pas formatée, si elle est protégée, ou si le directory est absent (disquettes "fichiers" de PFS File par exemple), un message adéquat est affiché.

- le type de SED
- le nom ou "Pathname" (s'il s'agit de ProDOS ou p-UCSD), ou le numéro (DOS 3.3)
- la place libre en nombre de secteurs (DOS 3.3), en blocs (ProDOS et p-UCSD) ou en Koctets (CP/M)

- la place qu'il occupe (mêmes unités que ci-dessus)
- son nom
- son type (text., bas., code., ...).

Pourquoi ce programme

C'est ainsi que j'ai été amené à rédiger ce Catalogue Multi-SED. Pour permettre une vision plus synthétique, l'affichage se fait sur trois colonnes et dans l'ordre alphabétique des noms des fichiers. Il permet aussi d'imprimer ce catalogue. Avec un petit effort supplémentaire, il devrait être possible de l'accoler à un "Catalogue général", comme celui paru dans Pom's 14, mais ça, je le laisse à votre soin.

les propriétés communes à chacun d'eux. Par exemple, j'ai renoncé à prendre en compte la date de création, que l'on ne trouve pas sous DOS 3.3, ni sous CP/M.

Pour comprendre le fonctionnement de ce programme, il faut savoir que depuis 1980, le formatage utilisé par Apple sur toute disquette 5 pouces 1/4, quel que soit le SED, comprend 35 pistes concentriques, divisées chacune en 16 secteurs (un secteur permet le stockage de l'équivalent de 256 octets). La reconnaissance de ces secteurs dépend non pas du SED, mais de la carte contrôleur des drives. C'est pourquoi il est possible de lire un secteur appartenant à une disquette formatée sous n'importe quel SED à partir d'un autre SED, en particulier du DOS 3.3 : c'est ce que réalise ce programme (qui aurait pu être écrit tout aussi bien sous ProDOS, p-UCSD ou CP/M 80). Il existe une routine en langage machine sous DOS 3.3, nommée RWTS (Read Write Track Sector). Par une trentaine de POKE, on transmet les informations nécessaires à cette routine, telles que numéro du drive, piste, secteur, adresse en mémoire où recopier ce secteur, ..., puis un CALL fait le reste (voir le Manuel du DOS).

DOS 3.3	0 1 2 3			32 33 34																											
	e/t	Nom du fichier (30 caractères)																												lo hi	
ProDOS	0 1	16 17														20 21				38											
	E	Nom du fichier (15 caractères)														T	lo hi														
p-UCSD	0 1 2 3 4				18 19														25												
	ad ar		T	E	Nom du fichier (15 caractères)																										
CP/M 80	0 1	9 10 11								15				31																	
	E	Nom du fichier (8 car)								T	L																				

à l'autre : ProDOS et p-UCSD les utilisent par deux pour former un bloc, tandis que CP/M, d'une certaine manière, les coupe en deux.

Enfin, tout le monde sait que le SED a besoin d'une *Table des matières* contenant la liste des fichiers présents sur la disquette, leur type, leur longueur, leur adresse, ... L'ensemble de ces informations est regroupé dans le "directory ou catalog", pour lequel un certain nombre de secteurs sont réservés sur la disquette. C'est à ces secteurs qu'il faut s'intéresser.

- Pour DOS 3.3, il s'agit des secteurs 15 à 0 de la piste 17
- Pour ProDOS et p-UCSD : secteurs de 11 à 2 (blocs 2-5) de la piste 0
- Pour CP/M : secteurs 0, 6, 12, 3, 9, 15 de la piste 4.

Structure du directory

Caractéristiques du volume

Sous ProDOS et p-UCSD, on trouve le nom du volume dans les premiers octets du premier secteur. Sous p-UCSD, on y trouve également le nombre de fichiers inscrits.

Sous DOS 3.3, le numéro du volume doit être recherché au début de la VTOC, c'est-à-dire le secteur 0 (en fait, il est aussi inscrit entre chaque secteur, ce qui permet à RWTS de lire ce numéro à n'importe quel endroit de la disquette).

Caractéristiques des fichiers

Les informations relatives à un fichier donné sont inscrites dans un certain ordre à l'intérieur d'un champ d'une longueur variant de 26 (p-UCSD) à 39 (ProDOS). La structure de ce champ, bien évidemment spécifique à chaque SED, est la suivante (seulement les caractéristiques utilisées dans le programme) :

voir schéma page 17

Légende

E : Etat indique si le fichier est effacé ou non, ainsi que pour ProDOS et p-UCSD, la longueur du nom du fichier, 1 octet.

T : Type du fichier (text, basic, code, binaire,...), 1 octet sauf CP/M qui a 3 octets.

LoHi : Longueur du fichier sous DOS 3.3 et ProDOS, 2 octets.

ADAF : Adresse Début et Adresse Fin du fichier sous p-UCSD, la longueur est obtenue par différence, 2 octets.

L : Longueur du fichier sous CP/M, 1 octet.

NOM : Nom du fichier, tel qu'il a été fourni par l'utilisateur lors de la sauvegarde, 8 à 30 octets selon le SED.

Ces champs sont inscrits séquentiellement sur le premier secteur réservé

au "directory", éventuellement après les caractéristiques du volume, puis sur le deuxième secteur, le troisième, et ainsi de suite. Le nombre maximum de fichiers dépend donc du nombre de secteurs réservés et de la longueur du champ : il est de 105 pour DOS 3.3, 51 pour ProDOS, 77 pour p-UCSD et 48 pour CP/M.

Il est une possibilité intéressante de ProDOS que j'ai longtemps ignorée, c'est celle de "sous-directory" : une excuse à ma paresse est que son utilisation est surtout justifiée avec un disque dur. Néanmoins, il serait assez facile d'inclure dans le programme la lecture de ces sous-directory (avis aux amateurs).

Description du programme

Une fois le type de SED identifié, le principe général est de recopier en mémoire tous les secteurs réservés au "directory" à partir de l'adresse 28416 (\$6F00), d'où le HIMEM:28416. On utilise pour cela la routine RWTS, dont les vecteurs sont POKES aux adresses 768 et suivantes.

C'est volontairement que j'utilise une adresse relativement basse pour la copie (28416), car cela permet, en cas de compilation, de déclarer comme entières toutes les variables utilisées ultérieurement pour aller PEEKer les informations (on reste en-dessous du fatal 32768).

Menu

En principe, il suffit de presser RETURN et le programme détermine tout seul, comme un grand, le type de SED. Les routines d'identification ont été testées sur des dizaines et des dizaines de disquettes, sans jamais les prendre en défaut, la possibilité de choix est donc laissée à l'utilisateur. Si celui-ci choisit un SED par un numéro de 1 à 4, une routine vérifie si ce choix est le bon, si oui elle renvoie aux lignes spécifiques de celui-ci, sinon elle le détermine.

Vérification

Un seul secteur est lu, puis un certain nombre d'octets d'identification sont comparés (lignes 320-410). Ces octets sont les suivants :

- DOS 3.3 (sect.0, piste 17) valeur de 3-0-0 pour les octets de rang 3-4-5
- ProDOS (sect.11, piste 0) valeur de 0-0-3 pour les octets de rang 0-1-2
- p-UCSD (sect.11, piste 0) valeur de 0-0-6 pour les octets de rang 0-1-2
- CP/M (sect.0, piste 4) les octets de rang 1-8 doivent correspondre au code de caractères imprimables

(ASCII 31-128) et les octets 13-14 doivent être nuls.

L'identification de CP/M est sans aucun doute la plus hasardeuse, mais je n'ai rien trouvé d'autre. C'est d'ailleurs pour cette raison qu'elle figure en dernière place : c'est presque une identification par élimination.

Lecture et écriture

Une fois le SED identifié, tous les secteurs réservés pour le "directory" sont systématiquement recopiés en mémoire vive, bien que souvent les derniers soient vides (en DOS 3.3, le dernier secteur comprend les fichiers 91 à 105). Cette opération est réalisée secteur par secteur en initialisant les vecteurs spécifiques du SED, soient : la piste, le secteur et l'adresse mémoire, puis en lançant RWTS.

Il faut noter que pour les trois premiers SED, les secteurs sont contigus : il suffit de donner le numéro du premier (PS) et du dernier (DS), et de les placer dans une boucle (ligne 470). Pour CP/M, ces secteurs ne sont pas contigus, et ne correspondent à aucune progression logique simple, d'où l'utilisation de DATA (lignes 480 et 1200).

A chaque lecture successive d'un secteur, on ajoute 256 (=longueur d'un secteur) à l'adresse de début de mémoire pour la copie (PA, incrémenté au début par DE).

Recherche du type de SED

On essaie successivement la routine de vérification pour chaque SED. Si les octets comparés correspondent à l'un d'eux, tout est bien, sinon on affiche un "SED INCONNU" et le programme recommence au début. A noter que si la routine rencontre quelque chose d'inhabituel, par exemple qu'elle ne trouve pas la piste et le secteur, la valeur de 64 (\$40) est retournée dans la table (ligne 440). Il s'agit généralement d'un disque non formaté ou protégé par un SED spécial.

Lecture des différents SED

Les variables utilisées sont les suivantes :

NV\$ Nom ou numéro du volume
 NF Nombre de fichiers
 UT Nombre d'unités (secteurs ou blocs) utilisées
 PA Nombre d'octets utilisés pour un fichier
 LI Nombre d'unités libres
 TY Code du type de fichier
 DIV Nombre de fichiers maximum sur une unité (secteur ou bloc)
 NOS() Type des fichiers
 MAX Nombre maximum d'unités disponibles
 LF Longueur du fichier
 N\$ Nom du fichier
 N\$() STR\$(LF)+N\$+NOS()

DOS 3.3

Ligne 570 : Lecture du numéro du volume; celui-ci est retourné par RWTS, il n'y a qu'à le lire.

Ligne 580 : Initialisation des variables, en particulier des types. A signaler l'existence de "typB" et "typA", qui normalement apparaissent sous B(inaire) : MULTIPLAN par exemple sauve ses fichiers sous la forme typB.

Ligne 590 : Début de la boucle, on lit les informations recopiées en mémoire en commençant à DE (28416). Un test est nécessaire à chaque fois que l'on recommence un nouveau secteur, pour repositionner les octets décisifs, soit tous les 7 fichiers.

Ligne 600 : Si les octets réservés au Nom du fichier sont nuls, on est arrivé au bout du "directory".

Ligne 610 : Si le premier octet vaut 255, le fichier est effacé.

Ligne 620 : Lecture de la longueur du fichier, et formatage pour l'écriture. On ne prend ici qu'un seul octet, ce qui pourrait causer une erreur si un fichier était d'une longueur supérieure à 255 (\$FF) secteurs. En fait, la commande CATALOG du DOS ne fait pas mieux, car où charger un fichier d'une telle longueur (plus de 65000 octets) ?

Ligne 630 : On additionne les secteurs utilisés.

Lignes 640-650 : Lecture des caractères formant le Nom du fichier. Bien que le nom puisse comporter jusqu'à 30 caractères, on ne prend que les 16 premiers (faut pas exagérer). Sous DOS 3.3, les octets composant le Nom du fichier ont leur bit de poids fort à un, il faut donc soustraire la valeur 128 avant de convertir en caractère ASCII. Comme certains modifient ces octets pour agrémenter leur "catalog" avec des CTRL-H ou des caractères inversés, j'ai inclus un test pour remplacer tous ces caractères anormaux par un "?".

Ligne 660 : Lecture du code du type du fichier.

Ligne 670 : On incrémente de 1 le nombre de fichiers existants.

Ligne 680 : On incrémente de 35 (=longueur d'un champs consacré à un fichier), puis retour de la boucle.

Ligne 690 : Calcul du nombre de secteurs libres, puis on passe la main aux routines d'affichage.

ProDOS

La logique est la même que ci-dessus, avec les différences suivantes :

Ligne 710 : Lecture du nom du volume (=Pathname), en commençant par sa longueur, exprimée sous un format un peu spécial, qui nous oblige à une petite gymnastique.

Ligne 720 : Cette fois, c'est vraiment

la lecture du Nom du volume.

Ligne 800 : Lorsqu'un nom de fichier est réécrit sur un ancien, seuls les nouveaux caractères sont inscrits. Remplacer "FICHIERNUMERO1" PAR "FICHER2" aboutira, sur la disquette à "FICHER2UMERO1". Mais si vous faites un CATALOG, vous ne verrez que "FICHER2". Ceci parce que l'octet qui précède le nom donne la longueur de celui-ci, ce qui permet au CATALOG de n'afficher que le nombre de caractères indiqués : lors de la première sauvegarde, cette longueur était de 14, elle n'est plus que de 8 lors de la deuxième. Cette valeur est contenue dans la variable LN.

Ligne 830 : Si l'on rencontre un fichier de type "DIR", il peut y avoir des fichiers présents sur la disquette, mais absents du directory principal. Dès lors, on est obligé, pour connaître le nombre réel de blocs utilisés, de lire le Volume Bit Map (équivalent de la VTOC du DOS 3.3). On place donc un drapeau (SD=1) pour faire ce travail, seulement si nécessaire, une fois la lecture du directory terminée.

Lignes 850-900 : Le décryptage du Volume Bit Map (bloc 6) m'a posé quelques problèmes, et pourtant c'est assez simple. En effet, si l'on sait que seuls les 35 premiers octets sont utilisés (soit exactement le nombre de pistes), et qu'un octet comporte 8 bits (soit tout aussi exactement le nombre de blocs par piste), la solution est toute trouvée (mais que d'essais dans le vide avant d'arriver là !). Ainsi, par exemple, si le bit 3 de l'octet 0 est à 1, on en conclut que le bloc 3 de la piste 0 est libre. En particulier, si un octet vaut 255 (=1 + 2 + 4 + 8 + 16 + 32 + 64 + 128), la piste correspondante est libre, soit 8 blocs. Si au contraire il vaut 0, alors toute la piste est allouée. Enfin, s'il est compris entre ces deux valeurs, un peu de gymnastique est nécessaire.

Parmi les types de fichiers, ceux utilisés par Appleworks (ADP,AWP et ASP) sont reconnus, mais il peut y en avoir d'autres, créés par des progiciels ProDOS que je ne connais pas, et qui ne seront pas reconnus : ils seront affichés par "....".

Pascal-UCSD

L'organisation de p-UCSD est assez comparable à ProDOS : unités en blocs, indication de la longueur (LN) du nom du volume ou du fichier,...

Ligne 930 : le nombre de fichiers (NF) présents sur la disquette est inscrit au début du premier bloc : ça facilite bien la suite.

Ligne 950 : En lieu et place de la longueur, on connaît l'adresse du premier bloc et celle du dernier. Mais comme sous UCSD les fichiers sont

sauvés sur des blocs contigus (pas d'entrelaçage), la différence nous donne la longueur effective.

CP/M 80

La particularité de CP/M est de n'accepter que des fichiers d'une longueur inférieure ou égale à 16 demi-secteurs. Si un fichier dépasse cette valeur, il est écrit sur le "directory" autant de fois qu'il comporte de blocs ou de fraction de bloc de 16 demi-secteurs. On trouve quand même tout ce qu'il faut pour établir un lien éventuel (si PEEK(PA+12)=0) entre de tels fichiers. C'est ce que réalisent les tests des lignes 1140-1150, et la routine des lignes 1040-1050. Le résultat est que le nombre d'unités (demi-secteurs) est compté chaque fois, le nombre de fichiers une seule fois, et le Nom du fichier n'apparaîtra qu'une fois également. C'est un peu long, mais on y arrive.

Ligne 1060 : Initialisation des variables : remarquer qu'il n'y a pas de code pour le type de fichier, celui-ci étant inscrit en clair (ASCII) à la suite du nom du fichier.

Ligne 1100 : Les unités sont converties en Koctets, comme le fait CP/M; le décompte peut ne pas être tout à fait exact, mais il correspond exactement à la valeur retournée par la fonction CP/M "STAT".

Tri et édition

Ouf ! Le reste est plus classique et ne nécessite pas d'explications particulières. Un point seulement, l'ordre d'affichage se fait de haut en bas sur la première colonne, puis de haut en bas sur la deuxième, et la suite sur la troisième.

Bibliographie

Pour comprendre la structure des différents SED, j'ai procédé par tâtonnement à l'aide d'un éditeur de secteurs. J'ai heureusement pu m'aider en consultant quelques ouvrages, très riches pour DOS 3.3, beaucoup moins pour p-UCSD. Pour ProDOS, la documentation est encore maigre. Quant à CP/M, j'ai dû tout chercher par essais et lecture.

DOS 3.3

Beneath Apple DOS, Don Worth & Pieter Leschner, Quality Software, 1981

P-UCSD

Un catalogue général en Pascal, Michel Crimont, Pom's 2

Gestion de fichiers et de périphériques pour AppleII/Pascal, Hervé Haut, Editions PSI, 1983

ProDOS

ProDOS Technical Reference Manual, Apple Computer, Inc., 1983 (assez sommaire sur bien des points).

Programme MULTISED.BAS

```
50 REM A.GOGNIAT, FEV. 1985
80 HIMEM: 28416
90 CL80 = 3: REM
```

NB :

La valeur de la variable 'CL80' doit être mise à :

*<1> si Apple //e+80 COL,
<2> si seulement 80 COL,
<3> si seulement 40 COL.*

```
100 DIM N$(105), CO$(255)
110 DEF FN MOD(A) = (A / DIV) - INT(A / DIV): ADR =
    768: DE = 28416
120 FOR I = AD TO AD + 29: READ BI: POKE I, BI: NEXT :
    REM MISE EN PLACE DE RWTS
130 DATA
    169,3,160,9,32,217,3,96,0,1,96,1,0,17,15,26,3,0,
    111,0,0,1,0,0,96,1,0,1,239,216
```

140 REM **MENU**

```
150 TEXT : GOSUB 1970
160 SD$(0) = "NE SAIT PAS": SD$(1) = "DOS 3.3": SD$(2)
    = "PRODOS 1.01": SD$(3) = "PASCAL UCSD": SD$(4)
    = "CPM 80": SD$(5) = "SORTIE"
170 PRINT : INVERSE : VTAB (1): PRINT "CATALOGUE
    POUR 4 SYSTEMES D'EXPLOITATION": VTAB (2):
    PRINT "   AFFICHAGE SUR 3 COLONNES   ":
    NORMAL : POKE 34,3
180 VTAB (7): PRINT "QUEL TYPE DE DOS?": PRINT :
    FOR I = 0 TO 5: PRINT : HTAB (10): INVERSE :
    PRINT I: NORMAL : PRINT " - "; SD$(I): NEXT
190 VTAB (23): HTAB (10): PRINT "---->": GET
    AS: CHOIX = VAL (AS): IF CHOIX > 5 THEN PRINT
    CHR$(7): GOTO 190
200 IF CHOIX = 0 THEN CHOIX = 6
210 ON CHOIX GOTO 220,240,260,280,1870,500
220 GOSUB 320: IF TY < > CH THEN 1840
230 PRINT " "SD$(CH): PIST = 17: PS = 15: DS = 0:
    GOSUB 470: GOSUB 300: GOTO 570: REM DOS
    3.3
240 GOSUB 340: IF TY < > CH THEN 1840
250 PRINT " "SD$(CH): PIST = 0: PS = 11: DS = 3: GOSUB
    470: GOSUB 300: GOTO 710: REM PRODOS
260 GOSUB 360: IF TY < > CH THEN 1840
270 PRINT " "SD$(CH): PIST = 0: PS = 11: DS = 5: GOSUB
    470: GOSUB 300: GOTO 920: REM P-UCSD
280 GOSUB 380: IF TY < > CH THEN 1840
290 PRINT " "SD$(CH): PIST = 3: FOR I = 1 TO 6: READ
    SE(I): NEXT : GOSUB 480: GOSUB 300: GOTO
    1030: REM CPM
300 HOME : VTAB (10): PRINT "S.E.D. RECONNU :
    "; SD$(CH): VTAB (16): PRINT "NOMBRE DE
    FICHIERS: "; RETURN
```

310 REM **VERIFICATION DU TYPE DE DOS**

```
320 PIST = 17: SE = 0: GOSUB 430: IF PEEK (DE + 3) = 3
```

```
AND PEEK (DE + 4) = 0 AND PEEK (DE + 5) = 0
THEN TY = 1: REM DOS 3.3
```

```
330 RETURN
```

```
340 PIST = 0: SE = 11: GOSUB 430: IF PEEK (DE) = 0
AND PEEK (DE + 1) = 0 AND PEEK (DE + 2) = 3
THEN TY = 2: REM PRODOS
```

```
350 RETURN
```

```
360 PIST = 0: SE = 11: GOSUB 430: IF PEEK (DE) = 0
AND PEEK (DE + 1) = 0 AND PEEK (DE + 2) = 6
THEN TY = 3: REM P-UCSD
```

```
370 RETURN
```

```
380 PIST = 3: SE = 0: GOSUB 430: IF PEEK (DE) < > 0
AND PEEK (DE) < > 31 AND PEEK (DE) < > 229
THEN 420: REM CPM
```

```
390 FOR I = 1 TO 8: IF PEEK (DE + I) < 31 OR PEEK (DE
+ I) > 128 THEN I = 8: GOTO 420
```

```
400 NEXT : FOR I = 13 TO 14: IF PEEK (DE + I) < > 0
THEN 420
```

```
410 NEXT : TY = 4
```

```
420 RETURN
```

```
430 POKE 781, PIST: POKE 782, SE: POKE 786, INT (DE
/ 256): CALL AD: REM MISE EN PLACE DES
VECTEURS RWTS SPECIFIQUES AU SED ET
LANCEMENT DE RWTS
```

```
440 IF PEEK (AD + 22) = 64 THEN 1880: REM I/O
ERREUR, DISQUE NON FORMATTE OU PROTEGE
```

```
450 RETURN
```

460 REM **LECTURE ET ECRITURE EN
MEMOIRE DES SECTEURS CONCERNES**

```
470 POKE AD + 13, PIST: PA = DE: FOR SE = PS TO DS
STEP - 1: GOSUB
```

```
490 : CALL AD: PA = PA + 256: NEXT : RETURN : REM
DOS 3.3, PRODOS ET PASCAL
```

```
480 POKE AD + 13, PIST: PA = DE: FOR I = 1 TO 6: SE =
SE(I): GOSUB
```

```
490 : CALL AD: PA = PA + 256: NEXT : RETURN : REM
CPM
```

```
490 POKE AD + 14, SE: POKE AD + 18, INT (PA / 256):
RETURN : REM RWTS, NUMERO SECTEUR A LIRE
ET ADRESSE MEMOIRE POUR COPIE
```

500 REM **RECHERCHE DU TYPE DE DOS**

```
510 GOSUB 320: IF TY = 1 THEN CH = 1: GOTO 230:
REM DOS 3.3 RECONNU
```

```
520 GOSUB 340: IF TY = 2 THEN CH = 2: GOTO 250:
REM PRODOS RECONNU
```

```
530 GOSUB 360: IF TY = 3 THEN CH = 3: GOTO 270:
REM P-UCSD RECONNU
```

```
540 GOSUB 380: IF TY = 4 THEN CH = 4: GOTO 290:
REM CPM RECONNU
```

```
550 VTAB (23): HTAB (1): PRINT "SYSTEME
D'EXPLOITATION INCONNU": GOTO 1850
```

560 REM **DOS 3.3**

```
570 NV$ = STR$ (PEEK (AD + 23)): PRINT : VTAB (13):
PRINT "VOLUME           ": NV$: VTAB (16)
```

```
580 NF = 0: UT = 0: PA = DE: GOSUB 1230
```

```

590 DIV = 7: FOR K = 1 TO 105: IF FN MOD(K - 1) = 0
    THEN PA = DE + 11 + (INT(K / 7) * 256): REM ON
    RECOMMENCE UN NOUVEAU SECTEUR
600 IF PEEK(PA + 3) = 0 THEN K = 105: GOTO 680: REM
    FIN DU CATALOG
610 IF PEEK(PA) = 255 THEN 680: REM FICHIER
    EFFACE
620 LF = PEEK(PA + 33): N$ = STR$(LF) + " ": FOR J =
    LEN(N$) TO 4: N$ = " " + N$: NEXT
630 UT = UT + LF
640 FOR J = PA + 3 TO PA + 18: CAR = PEEK(J): IF CAR
    = 136 OR CAR = 138 THEN CAR = 63
650 N$ = N$ + CHR$(CAR - 128 * (CAR > 127) + 96 *
    (CAR < 31)): NEXT: N$ = N$ + " "
660 TY = PEEK(PA + 2): TY = TY - (128 * (TY > 127)): CO$
    = CO$(TY)
670 NF = NF + 1: HTAB(22): PRINT NF: N$(NF) = N$ +
    CO$
680 PA = PA + 35: NEXT
690 LI = MAX - UT: GOTO 1320

```

700 REM PRODOS

```

710 LN = PEEK(DE + 4): LN = 16 * ((LN / 16) - INT(LN /
    16)): NV$ = ""
720 FOR I = 1 TO LN: NV$ = NV$ + CHR$(PEEK(DE + 4 +
    I)): NEXT: NV$ = " " + NV$: PRINT: VTAB(13):
    PRINT "VOLUME": "; NV$: VTAB(16)
730 NF = 0: PA = DE + 4: UT = 0: GOSUB 1240
740 FOR K = 2 TO 51
750 DIV = 13: IF FN MOD(K - 1) = 0 THEN PA = DE + 4 +
    ((K - 1) / 13) * 512: REM ON RECOMMENCE UN
    NOUVEAU BLOC
760 PA = PA + 39: LN = PEEK(PA): IF LN = 0 THEN 840:
    REM FICHIER EFFACE
770 IF PEEK(PA + 1) = 0 THEN K = 51: GOTO 840: REM
    FIN DE CATALOGUE
780 NF = NF + 1: HTAB(22): PRINT NF: LF = 256 * PEEK
    (PA + 20) + PEEK(PA + 19): UT = UT + LF
790 N$ = STR$(LF) + " ": FOR J = LEN(N$) TO 4: N$ = " "
    + N$: NEXT
800 DIV = 16: LN = 16 * FN MOD(LN)
810 FOR J = PA + 1 TO PA + LN: N$ = N$ + CHR$(PEEK
    (J)): NEXT: FOR J = LN TO 15: N$ = N$ + " ": NEXT
820 TY = PEEK(PA + 16): CO$ = CO$(TY): IF CO$ = ""
    THEN CO$ = "..."
830 N$(NF) = N$ + CO$: IF TY = 15 THEN SD = 1: REM
    ATTENTION, CE FICHIER SOUS-DIRECTORY
    BOULEVERSE NOTRE CALCUL DU NOMBRE DE
    BLOCS LIBRE, --> PASSER PAR ROUTINE 850-890
840 NEXT: IF SD = 0 THEN LI = MAX - UT: GOTO 1320
850 LI = 0: BMAP = DE + 1 + (8 * 256): FOR K = BMAP TO
    BMAP + 35: OC = PEEK(K): IF OC = 0 THEN 900
860 IF OC = 255 THEN LI = LI + 8: GOTO 900
870 IC = 256: FOR I = 1 TO 8: IC = IC / 2: IF OC > = IC
    THEN OC = OC - IC: LI = LI + 1
880 IF OC = 0 THEN I = 8
890 NEXT I
900 NEXT K: GOTO 1320

```

910 REM PASCAL UCSD

```

920 LN = PEEK(DE + 6): FOR I = 1 TO LN: NV$ = NV$ +

```

```

    CHR$(PEEK(DE + 6 + I)): NEXT: PRINT: VTAB
    (13): PRINT "VOLUME": "; NV$: VTAB(16)
930 NF = PEEK(DE + 16)
940 PA = DE: UT = 0: GOSUB 1260
950 FOR K = 1 TO NF: HTAB(22): PRINT K: PA = PA +
    26: LF = PEEK(PA + 2) - PEEK(PA): LF = LF + 256 *
    (LF < 0): UT = UT + LF
960 N$ = STR$(LF) + " ": FOR I = LEN(N$) TO 4: N$ = " "
    + N$: NEXT
970 LN = PEEK(PA + 6): FOR J = PA + 7 TO PA + 6 +
    LN: N$ = N$ + CHR$(PEEK(J)): NEXT
980 FOR J = LN TO 15: N$ = N$ + " ": NEXT
990 TY = PEEK(PA + 4): CO$ = CO$(TY)
1000 N$(K) = N$ + CO$
1010 NEXT: LI = MAX - UT: GOTO 1320

```

1020 REM CPM 80

```

1030 GOTO 1060
1040 LF = LF + VAL(LEFT$(N$(IN(L)), 4)): LF$ = STR$
    (LF): FOR J = LEN(LF$) TO 3: LF$ = " " + LF$: NEXT
1050 N$(IN(L)) = LF$ + " " + RIGHT$(N$(IN(L)), 19): L = II:
    GOTO 1190
1060 NF = 0: PA = DE: UT = 0: II = 0: GOSUB 1270
1070 FOR K = 1 TO 48
1080 IF PEEK(PA + 1) = 0 OR PEEK(PA + 1) = 229
    THEN K = 48: GOTO 1190: REM FIN DU CATALOG
1090 IF PEEK(PA) = 229 THEN 1190: REM FICHIER
    EFFACE
1100 DIV = 8: LF = PEEK(PA + 15): LF = INT(LF / 8) + (
    FN MOD(LF) > 0): UT = UT + LF
1110 N$ = STR$(LF) + " ": FOR J = LEN(N$) TO 4: N$ = " "
    + N$: NEXT
1120 FOR J = PA + 1 TO PA + 8: N$ = N$ + CHR$(PEEK
    (J)): NEXT: N$ = N$ + " "
1130 FOR J = PA + 9 TO PA + 11: N$ = N$ + CHR$(PEEK
    (J)): NEXT: N$ = N$ + " "
1140 IF LF < 16 AND PEEK(PA + 12) = 0 THEN 1180
1150 IF LF = 16 AND PEEK(PA + 12) = 0 THEN II = II +
    1: IN(II) = NF + 1: GOTO 1180: REM ATTENTION,
    CE FICHIER CONTINUE PLUS LOIN
1160 IF PEEK(PA + 12) > 0 THEN FOR L = 1 TO II: IF
    RIGHT$(N$(L), 19) = RIGHT$(N$(IN(L)), 19) THEN
    1040: REM ATTENTION, CE FICHIER EST LA
    SUITE D'UN FICHIER DEJA ENREGISTRE
1170 NEXT
1180 NF = NF + 1: HTAB(22): PRINT NF: N$(NF) = N$
1190 PA = PA + 32: NEXT K: LI = MAX - UT: GOTO 1320
1200 DATA 0,6,12,3,9,15: REM SECTEURS
    RESERVES AU DIRECTORY
1210 END

```

1220 REM TYPES DE FICHIERS

```

1230 MAX = 496: UN$ = "SECTEURS": CO$(0) = "TXT"
    ": CO$(1) = "INT": CO$(2) = "BAS": CO$(4) = "BIN"
    ": CO$(8) = "typS": CO$(16) = "REL": CO$(32) =
    "typA": CO$(64) = "typB": RETURN: REM DOS 3.3
1240 MAX = 273: UN$ = "BLOCKS": CO$(0) = "$00"
    ": CO$(1) = "$01": CO$(2) = "$02": CO$(3) = "$03"
    ": CO$(4) = "TXT": CO$(5) = "$05": CO$(6) = "BIN"
    ": CO$(7) = "$07": CO$(8) = "$08": REM CF. LIGNE
    SUIVANTE

```

```

1250 CO$(12) = "BAS ":CO$(15) = "DIR ":CO$(25) = "ADB
      ":CO$(26) = "AWP ":CO$(27) = "ASP ":CO$(250) =
      "INT ":CO$(251) = "VAR ":CO$(252) = "BAS
      ":CO$(253) = "VAR ":CO$(254) = "REL ":CO$(255) =
      "SYS ":RETURN:REM PRODOS
1260 MAX = 274:UN$ = "BLOCKS":CO$(1) = "BAD
      ":CO$(2) = "CODE":CO$(3) = "TEXT":CO$(4) =
      "INFO":CO$(5) = "DATA":CO$(6) = "GRAF":CO$(7) =
      "FOTO":CO$(8) = "? ":RETURN:REM P-UCSD
1270 MAX = 138:UN$ = "K":RETURN:REM AVEC CPM,
      C'EST PLUS SIMPLE
1280 REM

1290 REM  MISE EN FORME ET EDITION
1300 REM  DU CATALOGUE
1310 REM  TRI DES FICHIERS SHELL-METZNER

1320 PRINT:VTAB(19):PRINT "TRI ALPHABETIQUE ";
1330 EC = NF
1340 PRINT ". ";EC = INT(EC/2):IF EC < 1 THEN 1440
1350 J = 1:K = NF - EC
1360 I = J
1370 PL = I + EC
1380 IF MID$(N$(I),6,21) < MID$(N$(PL),6,21) THEN
      1420
1390 N$ = N$(I):N$(I) = N$(PL):N$(PL) = N$
1400 I = I - EC:IF I < 1 THEN 1420
1410 GOTO 1370
1420 J = J + 1:IF J > K THEN 1340
1430 GOTO 1360

1440 REM      EDITION SUR ECRAN

1450 TEXT:HOME:ON CL80 GOSUB 1920,1920,1960
1460 TI$(1) = "SYSTEME D'EXPLOITATION: ":TI$(2) =
      TI$(1):TI$(3) = "S.E.D. ":
1470 PRINT:INVERSE:PRINT TI$(CL80):SD$(CH):IF
      CH < > 4 THEN PRINT "VOLUME: ":NV$
1480 NORMAL:IF CH = 4 THEN PRINT
1490 PRINT NF;" FICHIERS":PRINT SPC(8);LI;" ";UN$;"
      LIBRES":PRINT
1500 NL = INT(NF/3):RESTE = NF - 3 * NL
1510 VTAB(5):AE = 1:ON CL80 GOSUB
      1660,1660,1730:AE = 0
1520 PRINT:PRINT
1530 VTAB(23):HTAB(1):INPUT "PRESSEZ <RETURN>
      ---->":A$
1540 HOME:ON CL80 GOSUB 1940,1960,1960:VTAB
      (10):PRINT "VOULEZ-VOUS IMPRIMER CE
      CATALOGUE?":PRINT:PRINT:PRINT " (O/N)
      ---->":GET A$
1550 IF A$ = "o" OR A$ = "O" THEN PRINT CHR$(13) +
      CHR$(4)"PR#1":GOTO 1580
1560 GOTO 1860

1570 REM      EDITION SUR IMPRIMANTE

1580 ON CL GOSUB 1930,1960,1960
1590 PRINT SD$(CH):IF CH < > 4 THEN PRINT SPC(
      5)"VOLUME: ":NV$
1600 IF CH = 4 THEN PRINT

```

```

1610 PRINT NF;" FICHIERS":SPC(8);LI;" ";UN$;"
      LIBRES":PRINT
1620 GOSUB 1660
1630 PRINT:PRINT:PRINT:PRINT CHR$(4)"PR#3":
      GOSUB 1930
1640 GOTO 1860

1650 REM  AFFICHAGE OU IMPRESSION SUR 3
      COLONNES: CARTE 80COL OU
      IMPRIMANTE

1660 FOR I = 1 TO NL
1670 IF I = 18 THEN IF AE THEN VTAB(23):PRINT
      "<RETURN> pour la suite ---->":GET A$:PRINT
      CHR$(10):VTAB(21):CALL - 958
1680 PRINT N$(I):N$(I + NL + (RESTE > 0)):N$(I + 2 * NL
      + (RESTE > 0) + (RESTE > 1))
1690 NEXT:IF NF < 4 THEN 1720
1700 IF RESTE > 0 THEN PRINT N$(NL + 1);
1710 IF RESTE > 1 THEN PRINT N$(2 * NL + 2)
1720 RETURN

1730 REM  AFFICHAGE SUR 3 COLONNES
      (SANS CARTE 80COL)

1740 FOR I = 1 TO NL
1750 IF I = 18 THEN VTAB(23):PRINT "<RETURN> pour
      la suite ---->":GET A$:PRINT CHR$(10):VTAB
      (21):CALL - 958
1760 N$ = N$(I):PRINT MID$(N$,6,8) + "." + RIGHT$
      (N$,4);
1770 N$ = N$(I + NL + (RESTE > 0)):PRINT MID$
      (N$,5,8) + "." + RIGHT$(N$,4);
1780 N$ = N$(I + 2 * NL + (RESTE > 0) + (RESTE > 1)):
      PRINT MID$(N$,5,8) + "." + RIGHT$(N$,4)
1790 NEXT:IF NF < 4 THEN 1820
1800 IF RESTE > 0 THEN N$ = N$(NL + 1):PRINT MID$
      (N$,6,8) + "." + RIGHT$(N$,4);
1810 IF RESTE > 1 THEN N$ = N$(2 * NL + 2):PRINT
      MID$(N$,5,8) + "." + RIGHT$(N$,4)
1820 RETURN

1830 REM  ROUTINE DE FIN OU DE RECOM-
      MENCEMENT

1840 PRINT:VTAB(22):PRINT CHR$(7)"ERREUR! IL
      NE S'AGIT PAS DE ":SD$(CH)
1850 PRINT "PRESSEZ UNE TOUCHE ---->":GET A$
1860 GOSUB 1930:RUN 150:REM  ON RECOMMENCE
      UN NOUVEAU CATALOGUE
1870 PRINT:CHR$(21):HOME:PRINT "POUR
      RECOMMENCER, TAPER 'RUN':END
1880 PRINT:VTAB(22):PRINT CHR$(7)"I/O ERR:
      DISQUE NON INIT. OU PROTEGE":GOTO 1850
1920 PRINT CHR$(13) + CHR$(4)"PR#3":RETURN:
      REM  ACTIVATION DE LA CARTE
1930 PRINT:CHR$(21):RETURN:REM
      DESACTIVATION DE LA CARTE
1940 PRINT:CHR$(17):RETURN:REM  40 COLONNES
1950 PRINT:CHR$(18):RETURN:REM  80 COLONNES
1960 RETURN:REM  REMPLACE L. 1930-1950 SI
      L'APPLE N'EST PAS UN IIE OU S'IL N'Y A PAS DE
      CARTE 80 COL
1970 CALL - 936:RETURN:REM PLACE L. 1920 S'IL
      N'Y A PAS DE CARTE 80 COL

```


La méthode PERT

Serge Cattani

Ce programme a été inspiré par l'article de Xuan Tung Bui paru dans le numéro de Septembre - Octobre 1982 de Micro-Systèmes, auquel le lecteur pourra se reporter s'il désire avoir une description plus précise de la méthode PERT.

Cette méthode fut créée il y a quelques années pour l'établissement de plannings de travaux de constructions. PERT signifie Program Evaluation and Review Technique. Associée à la méthode du chemin critique, cette méthode est couramment employée pour résoudre les problèmes d'ordonnancement et d'organisation.

L'analyse du chemin critique permet :

- de détecter les tâches critiques, c'est-à-dire celles dont l'exécution ne peut être ni retardée, ni ralentie sans que la durée totale des travaux ne soit prolongée.
- de déterminer le temps optimal nécessaire aux travaux.
- d'établir un ordonnancement.

Afin d'utiliser ce programme au mieux, il est conseillé de préparer un tableau comprenant la liste des tâches relatives aux travaux à effectuer avec en face de chaque tâche :

- le temps de réalisation le plus probable
- la durée minimale
- la durée maximale.

Après la constitution du tableau, il faudra dessiner sous forme de graphe, l'ensemble des activités de manière ordonnée.

Ce graphe se composera de noeuds, c'est-à-dire des cercles numérotés représentant les événements. Chaque cercle sera relié à un autre cercle par un arc, représentant les activités ou tâches.

Chaque noeud doit porter un numéro différent; le point d'arrivée d'un arc doit correspondre à un numéro de noeud supérieur à celui du noeud de départ.

Le chemin défini est donc une suite continue d'arcs dont l'extrémité terminale de l'un est l'extrémité initiale de l'autre. Il faut s'assurer que tous les chemins du graphe ne sont pas interrompus. Si un chemin ne peut aboutir, il est indispensable de créer un arc fictif de valeur nulle liant les extrémités interrompues.

Le planning obtenu sur l'imprimante affichera (voir schéma) :

— en haut : les numéros de noeuds,

— en ordonnée : les activités,
— en abscisse : la durée.

Mode d'emploi

Les possesseurs d'Apple //e avec la carte Chat Mauve doivent copier les programmes PURPLESOFT* et PURPLESOFT sur la disquette contenant PERT. Avant toute utilisation de PERT, il faut faire BRUN PURPLESOFT.

Signalons que PURPLESOFT ne fonctionne pas sur un Apple //c.

Pour lancer PERT, faire RUN PERT si l'on a un Apple //e en 80 colonnes ou un //c, sinon faire RUN PERT40. On obtient alors le menu suivant :

— Chemin critique >
— PERT > permet d'entrer un graphe avant de le traiter selon l'une ou l'autre méthode; le programme de-

mande si l'on veut imprimer les résultats : il utilise des codes pour EPSON RX80; ceux-ci sont à modifier si vous avez une autre imprimante.

— Lecture d'un graphe existant : un exemple est fourni sur la disquette d'accompagnement (fichiers MAISON et MAISON.RESULT); voir l'exemple en encadré.

— Impression du planning : appelle le programme PERT.GRAPH qui ne fonctionne que sur Apple //e avec la carte Chat Mauve; le hardcopy est prévu pour imprimante EPSON RX80, à changer si vous avez une autre imprimante.

— Modification d'un graphe existant.
— Mode d'emploi : appelle le programme PERT.TEXT, qui lance à son tour PERT.DEMO; ce dernier ne fonctionne que sur Apple //e avec la carte Chat Mauve.

Tâches		Temps de réalisation des travaux		
		probable	optimiste	pessimiste
a	Etablissement du plan	15	9	21
b	Obtention du permis de construire	30	30	45
c	Signature des contrats	5	3	10
d	Transport du matériel au chantier	2	2	3
e	Installation eau et électricité	3	3	3
f	Fondations	10	8	15
g	Transport de terre pour le jardin	2	2	13
h	Murs	10	8	12
i	Toit	12	7	14
j	Décoration	10	8	15
k	Décoration intérieure	10	9	13

Liste des travaux

événement	Délai		écart
	au plus tôt	au plus tard	
1	0,00	0,00	0,00
2	15,29	15,29	0,00
3	49,45	49,45	0,00
4	55,74	55,74	0,00
6	58,74	58,74	0,00
5	58,74	58,74	0,00
7	58,74	81,11	22,38
8	69,42	69,42	0,00
9	79,26	79,26	0,00
10	90,64	90,64	0,00
11	100,41	100,41	0,00

Table

Exemple

La construction d'une maison

voir schéma 1

Réseau de construction

Les arcs allant du noeud 5 au noeud 6 et du noeud 5 au noeud 7 sont des arcs fictifs de longueur nulle.

Liste des travaux

voir encadré

Traitement du chemin critique :

1-Classement ordonné des événements (noeuds) :

1 --> 2 --> 3 --> 4 --> 6 --> 5 --> 7 --> 8 --> 9 --> 10 --> 11 --> fin

2- Evènements :

voir table

3- Activités et durées correspondantes :

voir encadré

4- Ordre du chemin critique :

1 --> 2 --> 3 --> 4 --> 5 --> 6 --> 8 --> 9 --> 10 --> 11 --> fin

avec une longueur estimée de : 100.414677

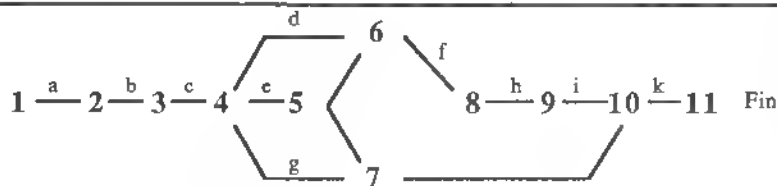


Schéma 1 : Réseau de construction

Activités		Temps attendu	Temps maximum	
1	de 1 à 2	15,29	15,29	*** arc critique ***
2	de 2 à 3	34,17	34,17	*** arc critique ***
3	de 3 à 4	6,28	6,28	*** arc critique ***
4	de 4 à 6	2,29	3,00	
5	de 4 à 5	3,00	3,00	*** arc critique ***
6	de 6 à 8	10,69	10,69	*** arc critique ***
7	de 4 à 7	2,25	25,38	
8	de 8 à 9	9,83	9,83	*** arc critique ***
9	de 9 à 10	11,39	11,39	*** arc critique ***
10	de 7 à 10	9,53	31,91	
11	de 10 à 11	9,77	9,77	*** arc critique ***
12	de 5 à 6	0,00	0,00	*** arc critique ***
13	de 5 à 7	0,00	22,38	

Activités et durées correspondantes

Programme PERT.GRAPH

```

40 & TEXT 1
50 PRINT
60 INVERSE : PRINT
70 HTAB 10: PRINT "GRAPHIQUE
  P.E.R.T. : NORMAL
80 HTAB 4: VTAB 23: PRINT "TOUCHE 'R'
  POUR RETOUR AU MENU"
90 ONERR GOTO 940
100 REM RECUP. RESULTATS
110 D$ = CHR$(4)
120 HTAB 5: VTAB 15: FLASH : PRINT
  "PREPAREZ L'IMPRIMANTE S.V.P."
130 NORMAL : HTAB 1: VTAB 8
140 INPUT "NOM DU FICHIER ? " : NE$
150 IF NE$ = "" THEN 130
160 IF NE$ = "R" OR NE$ = "r" THEN 860
170 REM RECUP. RESULTATS
180 PRINT D$;"OPEN":NE$;"RESULT"
190 PRINT D$;"READ":NE$;"RESULT"
200 INPUT A: INPUT N1
210 DIM N(N1),E(N1),L(N1),P(A),T(A),
  DD(A),S(A),LS(A)
220 FOR I = 1 TO N1
230 INPUT N(I): INPUT E(I): INPUT L(I)
240 NEXT
250 FOR I = 1 TO A
260 INPUT P(I): INPUT S(I): INPUT T(I):
  INPUT DD(I)
270 INPUT LS(I)
280 NEXT
290 INPUT N4

```

```

300 DIM CP(N4)
310 FOR I = 1 TO N4
320 INPUT CP(I)
330 NEXT
340 PRINT D$;"CLOSE"
350 REM GRAPHE PERT
360 & GR 9
370 & BACK = 0: & COLOR = 15
380 K = INT (220 / E(N1))
390 FOR I = 1 TO N1
400 IF E(I) = E(I - 1) AND I > 1 THEN L = 10
410 & POS K * E(I),182 : L
420 L = 0
430 & PRINT N(I):
440 & POS K * E(I),0
450 NEXT
460 FOR I = 1 TO N1
470 & PLOT K * E(I),10 TO K * E(I),180
480 NEXT
490 & PLOT 0,10 TO 560,10
500 REM ==AFFICHE ABSSISSE ==
510 FOR I = 0 TO L(N1) STEP 10: & POS K *
  I,0: & PRINT I: & PLOT K * I,6 TO K *
  I,14: NEXT
520 FOR I = 0 TO L(N1)
530 & PLOT K * I,8 TO K * I,12
540 NEXT
550 REM ----- DESSIN TACHES
560 K1 = INT (152 / A)
570 FOR I = 1 TO A
580 GOSUB 900
590 X = X * K : Y = 175 - K1 * I
600 & PLOT X,Y TO X + K * T(I),Y
610 IF INT (T(I) * 100 + .5) / 100 > = INT
  (DD(I) * 100 + .5) / 100 THEN Y = Y - 1
620 & PLOT X,Y TO X + K * T(I),Y
630 NEXT I
640 REM -----DESSIN POINTILLES
650 FOR I = 1 TO A

```

```

660 FOR J = 1 TO N4
670 IF S(I) = CP(J) AND T(I) < DD(I) THEN
  690
680 GOTO 740
690 GOSUB 900:X = K * X + K * T(I)
700 FOR JJ = X TO X + K * (DD(I) - T(I))
  STEP 3
710 & PLOT JJ,175 - K1 * I
720 NEXT JJ
730 GOTO 750
740 NEXT J
750 NEXT I
760 PRINT D$;"PR#1"
770 PRINT CHR$(9);"GE"
780 & GR 9
790 PRINT CHR$(12)
800 FOR I = 1 TO A
810 GOSUB 900
820 Y = 175 - K1 * I
830 & POS I,Y: & PRINT L$(I)
840 NEXT
850 PRINT CHR$(9);"GE"
860 & TEXT 5
870 PRINT D$;"PR#0": PRINT D$;"RUN
  PERT"
880 INPUT A$: PRINT ASC (A$)
890 REM ==RECHERCHE ABSISS
900 FOR J = 1 TO N1
910 IF N(J) = P(I) THEN X = E(J): GOTO 930
920 NEXT J
930 RETURN
940 REM TRAITEMENT ERREUR
950 IF PEEK (222) = 5 THEN VTAB 10:
  FLASH : PRINT "FICHIER INEXISTANT":
  NORMAL : GOTO 970
960 VTAB 10: FLASH : PRINT "< ERREUR
  >": NORMAL
970 FOR IT = 1 TO 1500: NEXT : RUN

```

Programme PERT

60 ONERR GOTO 2950

70 T\$ = "-----"

80 D\$ = CHR\$(4): PRINT D\$;"PR#3":PRINT
90 HOME

100 POKE 36,33: INVERSE : PRINT
"METHODE P.E.R.T.": NORMAL :
VTAB 8

110 PRINT
120 PRINT "CHEMIN CRITIQUE
-> C"

130 PRINT "P.E.R.T." -> P"
140 PRINT "LECTURE GRAPHE EXISTANT
-> L"

150 PRINT "IMPRESSION DU PLANNING
-> F"

160 PRINT "MODIF. GRAPHE EXISTANT
-> M"

170 PRINT "MODE D'EMPLOI
-> E"

180 PRINT "RETOUR AU BASIC
-> R"

190 POKE 36,33: VTAB 18: INPUT "VOTRE
CHOIX ? ";Q\$

200 IF Q\$ < "C" AND Q\$ < "P" AND Q\$ <
> "L" AND Q\$ < "M" AND Q\$ < "E"
AND Q\$ < "F" AND Q\$ < "R" AND Q\$
< "c" AND Q\$ < "p" AND Q\$ < "l"
AND Q\$ < "m" AND Q\$ < "e" AND Q\$
< "f" AND Q\$ < "r" THEN 90

210 IF Q\$ = "M" OR Q\$ = "m" THEN GOSUB
3000: GOTO 1740

220 IF Q\$ = "E" OR Q\$ = "e" THEN PRINT
D\$;"RUN PERT.TEXT"

230 IF Q\$ = "F" OR Q\$ = "f" THEN PRINT
D\$;"RUN PERT.GRAPH"

240 IF Q\$ = "R" OR Q\$ = "r" THEN END
250 VTAB 22: PRINT : INPUT

"VOULEZ-VOUS IMPRIMER LES
RESULTATS ? ";R\$

260 IF Q\$ < "L" AND Q\$ < "T" THEN 290
270 O = 1: HOME : INPUT "NOM DU
FICHIER ? ";NE\$

280 GOSUB 2820: GOTO 430
280 HOME : POKE 36,33: INVERSE : PRINT

"METHODE P.E.R.T.": NORMAL :
VTAB 8

300 PRINT : INPUT "NOMBRES
D'ACTIVITES (ARC) ? ";A\$

310 A = VAL (A\$): IF A < 2 THEN PRINT "
UN CHIFFRE > 1 S.V.P.": GOTO 300

320 GOSUB 340: GOTO 390
330 REM

340 IF ME = 1 THEN RETURN

350 DIM N(A + 1),E(A + 1),L(A + 1),LAG(A +
1),L\$(A + 1)

360 DIM
T1(A),T2(A),T3(A),T(A),SD(A),DD(A)
370 DIM CP(A + 1),KL(A + 1),P(A + 1),S(A +
1),R(A + 1)

380 ME = 1: RETURN

390 PRINT : FOR I = 1 TO A

400 PRINT : PRINT : INVERSE : PRINT
"ACTIVITE ";I: ": NORMAL

410 GOSUB 2500

420 NEXT I

430 PRINT : IF LEFT\$(R\$,1) = "O" OR
LEFT\$(R\$,1) = "o" THEN PRINT
D\$;"PR#1"

440 REM CLASSEMENT EVENEMENT

450 N1 = 0

460 FOR I = 1 TO A

470 FOR J = 1 TO N1

480 IF P(I) = N(J) GOTO 520

490 NEXT J

500 N1 = N1 + 1

510 N(N1) = P(I)

520 FOR J = 1 TO N1

530 IF S(I) = N(J) GOTO 570

540 NEXT J

550 N1 = N1 + 1

560 N(N1) = S(I)

570 NEXT I

580 IF O\$ = "C" OR Q\$ = "c" THEN 660

590 :

600 FOR I = 1 TO A

610 T(I) = (T2(I) + (4 * T1(I)) + T3(I)) / 6

620 SD(I) = (T3(I) - T2(I)) / 6

630 KI = 2 * RND (1) - 1

640 T(I) = T(I) + (SD(I) * KI)

650 NEXT I

660 PRINT : PRINT

670 REM

680 CL = 1: GOSUB 700: GOTO 910

690 REM ===== PROG DE CLASSEMENT

700 FOR I = 1 TO A

710 R(I) = I

720 NEXT I

730 A1 = A

740 A1 = A1 - 1

750 A2 = 0

760 FOR I = 1 TO A1

770 K = R(I)

780 K1 = R(I + 1)

790 IF CL = 1 THEN 820

800 IF S(K) > = S(K1) THEN 870

810 GOTO 830

820 IF P(K) < = P(K1) THEN 870

830 R1 = R(I)

840 R(I) = R(I + 1)

850 R(I + 1) = R1

860 A2 = 1

870 NEXT I

880 IF A2 = 1 THEN 740

890 RETURN

900 :

910 FOR I = 1 TO A

920 K = R(I)

930 A3 = P(K)

940 GOSUB 1750

950 I1 = K3

960 K = R(I)

970 A3 = S(K)

980 GOSUB 1750

990 I2 = K3

1000 K = R(I)

1010 M = E(I1) + T(K)

1020 IF E(I2) > = M THEN 1050

1030 K = R(I)

1040 E(I2) = E(I1) + T(K)

1050 NEXT I

1050 REM

1070 CL = 0: GOSUB 700

1080 :

1090 FOR I = 1 TO A

1100 K = R(I)

1110 A3 = S(K)

1120 GOSUB 1750

1130 I1 = K3

1140 K = R(I)

1150 A3 = P(K)

1160 GOSUB 1750

1170 I2 = K3

1180 K = R(I)

1190 M = L(I1) + T(K)

1200 IF L(I2) > = M THEN 1230

1210 K = R(I)

1220 L(I2) = L(I1) + T(K)

1230 NEXT I

1240 K = R(I)

1250 A3 = S(K)

1260 GOSUB 1750

1270 C = E(K3)

1280 FOR I = 1 TO N1

1290 L(I) = C - L(I)

1300 NEXT I

1310 REM CALCUL INTERVALLE TEMPS

1320 FOR I = 1 TO N1

1330 LAG(I) = L(I) - E(I)

1340 NEXT I

1350 REM

1360 FOR I = 1 TO N1

1370 R(I) = I

1380 NEXT

1390 FOR I = 1 TO N1 - 1: FOR J = I + 1 TO N1
- 1

1400 IF E(I) < = E(J) THEN 1430

1410 TA = E(I):E(I) = E(J):E(J) = TA

1420 TA = R(I):R(I) = R(J):R(J) = TA

1430 NEXT J,I

1440 REM RESULTATS

1450 GOSUB 1850

1460 REM CEHMIN CRITIQUE

1470 KK = 1

1480 FOR I = 1 TO N1

1490 IF ABS (LAG(I)) > = 0.01 THEN 1550

1500 LAG(I) = 0

1510 CP(KK) = N(I)

1520 KL(KK) = N(I)

1530 N4 = KK

1540 KK = KK + 1

1550 NEXT I

1560 N5 = N4 - 1

1570 FOR I = 1 TO N5

1580 I1 = I + 1

1590 FOR J = I1 TO N4

1600 IF KL(I1) > KL(J) THEN 1630

1610 IF KL(I1) < KL(J) THEN 1690

1620 IF CP(I1) < = CP(J) THEN 1690

1630 IT = KL(I1)

1640 JT = CP(I1)

1650 KL(I1) = KL(J)

1660 CP(I1) = CP(J)

1670 KL(J) = IT

1680 CP(J) = JT

1690 NEXT J,I

1700 GOSUB 2070

1710 IF LEFT\$(R\$,1) < "O" AND LEFT\$(
R\$,1) < "o" THEN VTAB 22: GOSUB
2690

1720 IF O = 1 OR Q\$ = "L" OR Q\$ = "l" THEN
1740

1730 GOSUB 2710

1740 PRINT D\$;"PR#0": RUN

1750 REM

1760 FOR J = 1 TO N1

1770 K3 = J

1780 IF CA = 1 THEN 1810

1790 IF N(K3) = A3 THEN 1840

1800 GOTO 1820

1810 IF N(R(K3)) = A3 THEN 1840

1820 NEXT J

1830 PRINT : PRINT "PAS DE NUMERO
D'ACTIVITE ";A3

1840 RETURN

1850 HOME

1860 PRINT CHR\$(14):" "NE\$: PRINT :
PRINT : PRINT

1870 PRINT "CLASSEMENT ORDONNE DES
EVENEMENTS (NOEUDS) :"

1880 PRINT "-----"

1890 PRINT :CE = 0

1900 FOR J = 1 TO N1: PRINT N(R(J)):"->":

1910 CE = CE + 1: IF CE > 13 THEN CE = 0:
PRINT

1920 NEXT J: PRINT "FIN": PRINT : PRINT :
PRINT

1930 IF LEFT\$(R\$,1) < "O" AND LEFT\$(
R\$,1) < "o" THEN GOSUB 2690

1940 REM IMPRESSION DELAIS

1950 HOME : POKE 36,35: INVERSE :

```

PRINT "EVENEMENTS ": NORMAL :
PRINT
1960 PRINT T$
1970 PRINT "      DELAI      "
1980 PRINT "EVENEMENT": POKE 36,15:
PRINT "AU PLUS TOT": POKE 36,40:
PRINT "AU PLUS TARD":
1990 POKE 36,60: PRINT "ECART"
2000 PRINT T$
2010 FOR I = 1 TO N1
2020 PRINT N(R(I)): POKE 36,19: PRINT
INT (100 * E(I) + .5) / 100: POKE 36,44:
PRINT INT (100 * L(R(I)) + .5) / 100:
2030 POKE 36,62: PRINT INT (100 *
LAG(R(I)) + .5) / 100
2040 NEXT I
2050 PRINT T$
2060 RETURN
2070 REM CHEMIN CRITIQUE
2080 IF LEFT$(R$,1) < > "O" AND LEFT$(
R$,1) < > "o" THEN GOSUB 2690
2090 PRINT CHR$(12)
2100 T5 = 0: PRINT : PRINT
2110 HOME : POKE 36,18: INVERSE :
PRINT " ACTIVITES ET DUREES
CORRESPONDANTES.": NORMAL :
PRINT
2120 PRINT T$
2130 PRINT "ACTIVITES": POKE 36,12:
PRINT "DE": POKE 36,20: PRINT "A":
POKE 36,24: PRINT "TEMPS
ATTENDU": POKE 36,40: PRINT
"TEMPS MAX."
2140 PRINT T$
2150 CA = 1
2160 FOR I = 1 TO A
2170 A3 = P(I)
2180 GOSUB 1750
2190 I1 = K3
2200 A3 = S(I)
2210 GOSUB 1750
2220 I2 = K3
2230 D = L(R(I2)) * E(I1)
2240 DD(I) = D
2250 IF ABS(T(I) - D) < 0.001 THEN 2290
2260 PRINT "A": POKE 36,12: PRINT
P(I): POKE 36,15: PRINT "->": POKE
36,20: PRINT S(I): POKE 36,28: PRINT
INT (100 * T(I) + .5) / 100:
2270 POKE 36,44: PRINT INT (100 * D + .5) /
100
2280 GOTO 2320
2290 PRINT "A": POKE 36,12: PRINT
P(I): POKE 36,15: PRINT "->": POKE
36,20: PRINT S(I): POKE 36,28: PRINT
INT (100 * T(I) + .5) / 100:
2300 POKE 36,44: PRINT INT (100 * D + .5) /
100: POKE 36,56: PRINT "ARC
CRITIQUE ""
2310 T5 = T5 + T(I)
2320 NEXT I
2330 PRINT T$
2340 IF LEFT$(R$,1) < > "O" AND LEFT$(
R$,1) < > "o" THEN GOSUB 2690
2350 HOME
2360 PRINT : PRINT
2370 PRINT "VOICI L'ORDRE DU CHEMIN
CRITIQUE : "
2380 PRINT "-----"
2390 FOR J = 1 TO N4: PRINT CP(J); "-> ":
2400 NEXT J
2410 PRINT "FIN"
2420 PRINT : PRINT
2430 PRINT "... AVEC UNE LONGUEUR
ESTIMEE DE :"; T5
2440 IF Q = 1 THEN 2480
2450 HTAB 0: VTAB 15
2460 INPUT "NOM DU FICHIER A

```

```

SAUVEGARDER ? "; NE$
2470 IF NE$ = "" THEN 2450
2480 GOSUB 3120: REM SAUVEGARDE
RESULTATS
2490 RETURN
2500 REM ENTREE DONNEES
2510 INPUT "LIBELLE DE L'OPERATION
": L$(I)
2520 INPUT "POINT DE DEPART :"; P$: P(I)
= VAL (P$)
2530 INPUT "POINT D'ARRIVEE :"; S$: S(I)
= VAL (S$)
2540 IF P(I) > = S(I) OR P(I) < = 0 THEN
PRINT "< ERREUR >": GOTO 2520
2550 IF Q$ < > "C" AND Q$ < > "c" THEN
2590
2560 INPUT "DUREE "; T1$
2570 IF T1$ < > "0" AND VAL (T1$) = 0 THEN
PRINT "< ERREUR >": GOTO 2560
2580 T(I) = VAL (T1$): GOTO 2670
2590 INPUT "DELAI D'EXECUTION LE PLUS
PROBABLE :"; T1$
2600 IF T1$ < > "0" AND VAL (T1$) = 0 THEN
PRINT "< ERREUR >": GOTO 2590
2610 T1(I) = VAL (T1$)
2620 IF T1(I) = 0 THEN T2(I) = 0: T3(I) = 0:
GOTO 2670
2630 INPUT "      LE PLUS OPTIMISTE
": T2$: T2(I) = VAL (T2$)
2640 IF T2(I) > T1(I) OR (T2$ < > "0" AND
VAL (T2$) = 0) THEN PRINT "< ERREUR
>": GOTO 2630
2650 INPUT "      LE PLUS PESSIMISTE
": T3$: T3(I) = VAL (T3$)
2660 IF T3(I) < T1(I) THEN PRINT "<
ERREUR >": GOTO 2650
2670 RETURN
2680 REM
2690 PRINT : PRINT "APPUYEZ SUR UNE
TOUCHE POUR LA SUITE.": GET C$
2700 RETURN
2710 REM SAUVEGARDE SUR DISQUE
2720 PRINT
2730 PRINT D$;"OPEN"; NE$
2740 PRINT D$;"WRITE"; NE$
2750 PRINT A: PRINT Q$
2760 FOR I = 1 TO A
2770 PRINT P(I): PRINT S(I): PRINT T(I):
PRINT T1(I): PRINT T2(I): PRINT T3(I)
2780 PRINT L$(I)
2790 NEXT
2800 PRINT D$;"CLOSE"
2810 RETURN
2820 REM LECTURE ENREGISTREMENT
2830 INVERSE : HTAB 20: VTAB 10: PRINT
"UN PEU DE PATIENCE...": NORMAL
2840 PRINT
2850 PRINT D$;"OPEN"; NE$
2860 PRINT D$;"READ"; NE$
2870 INPUT A: INPUT Q$
2880 A = A + 1: GOSUB 340: A = A - 1
2890 FOR I = 1 TO A
2900 INPUT P(I): INPUT S(I): INPUT T(I):
INPUT T1(I): INPUT T2(I): INPUT T3(I)
2910 INPUT L$(I)
2920 NEXT
2930 PRINT D$;"CLOSE"
2940 RETURN
2950 REM TRAITEMENT ERREUR
2960 IF PEEK (222) = 5 THEN VTAB 10:
PRINT "FICHIER INEXISTANT": GOTO
2980
2970 PRINT "ERREUR"
2980 FOR IT = 1 TO 1000: NEXT
2990 GOTO 1740
3000 REM MODIF ENREGISTREMENT
3010 HOME
3020 INPUT "NOM DU FICHIER A MODIFIER

```

```

? "; NE$
3030 GOSUB 2940
3040 PRINT "VOULEZ - VOUS AJOUTER
UNE ACTIVITE AUX "; A;" ACTIVITES
EXISTANTES ";
3050 INPUT R$
3060 IF LEFT$(R$,1) < > "O" AND LEFT$(
R$,1) < > "o" THEN 3080
3070 A = A + 1: A = A: GOTO 3090
3080 INPUT "NUMERO DE L'ACTIVITE A
MODIFIER ? "; I
3090 GOSUB 2510
3100 GOSUB 2720
3110 RETURN
3120 REM SAUVEGARDE RESULTATS
3130 PRINT D$;"OPEN"; NE$;"RESULT"
3140 PRINT D$;"WRITE"; NE$;"RESULT"
3150 PRINT A: PRINT N1
3160 FOR I = 1 TO N1
3170 PRINT N(R(I)): PRINT E(I): PRINT
L(R(I))
3180 NEXT
3190 FOR I = 1 TO A
3200 PRINT P(I): PRINT S(I): PRINT T(I):
PRINT DD(I)
3210 PRINT L$(I)
3220 NEXT
3230 PRINT N4
3240 FOR I = 1 TO N4
3250 PRINT CP(I)
3260 NEXT
3270 PRINT D$;"CLOSE"
3280 RETURN

```

Programme PERT40

```

60 ONERR GOTO 2960
70 T$ = "-----"
80 D$ = CHR$(4): PRINT
90 HOME
100 HTAB 10: INVERSE : PRINT "METHODE
P.E.R.T.": NORMAL : VTAB 8
110 PRINT
120 PRINT "CHEMIN CRITIQUE      -> C"
130 PRINT "P.E.R.T.              -> P"
140 PRINT "LECTURE GRAPHE EXISTANT
-> L"
150 PRINT "IMPRESSION DU PLANNING
-> I"
160 PRINT "MODIF. GRAPHE EXISTANT
-> M"
170 PRINT "MODE D'EMPLOI          -> E"
180 PRINT "RETOUR AU BASIC        -> R"
190 HTAB 10: VTAB 18: FLASH : PRINT
"VOTRE CHOIX ? ": NORMAL : GET Q$:
PRINT Q$: NORMAL
200 IF Q$ < > "C" AND Q$ < > "P" AND Q$ <
> "L" AND Q$ < > "M" AND Q$ < > "E"
AND Q$ < > "I" AND Q$ < > "R" AND Q$ <
> "c" AND Q$ < > "p" AND Q$ < > "l" AND
Q$ < > "m" AND Q$ < > "e" AND Q$ < >
"i" AND Q$ < > "r" THEN 90
210 IF Q$ = "M" OR Q$ = "m" THEN GOSUB
3010: GOTO 1740
220 IF Q$ = "E" OR Q$ = "e" THEN PRINT
D$;"RUN PERT.TEXT"
230 IF Q$ = "I" OR Q$ = "i" THEN PRINT
D$;"RUN PERT.GRAPH"
240 IF Q$ = "R" OR Q$ = "r" THEN END
250 VTAB 22: PRINT : INPUT
"VOULEZ-VOUS IMPRIMER LES
RESULTATS ? "; R$
260 IF Q$ < > "L" AND Q$ < > "l" THEN 290
270 Q = 1: HOME : INPUT "NOM DU FICHIER
? "; NE$

```

```

280 GOSUB 2830: GOTO 430
290 HOME : HTAB 10: INVERSE : PRINT
  "METHODE P.E.R.T.": NORMAL : VTAB
  8
300 PRINT : INPUT "NOMBRES
  D'ACTIVITES (ARC) ? "A$
310 A = VAL(A$): IF A < 2 THEN PRINT "UN
  CHIFFRE > 1 S.V.P.": GOTO 300
320 GOSUB 340: GOTO 390
330 REM
340 IF ME = 1 THEN RETURN
350 DIM N(A + 1), E(A + 1), L(A + 1), LAG(A +
  1), L$(A + 1)
360 DIM
  T1(A), T2(A), T3(A), T(A), SD(A), DD(A)
370 DIM CP(A + 1), KL(A + 1), P(A + 1), S(A +
  1), R(A + 1)
380 ME = 1: RETURN
390 PRINT : FOR I = 1 TO A
400 PRINT : PRINT : INVERSE : PRINT
  "ACTIVITE "; I; ": NORMAL
410 GOSUB 2510
  NB : Les lignes 420 à 1700 sont
  identiques à celles du
  programme PERT
1710 IF LEFT$(R$, 1) < > "O" AND LEFT$(
  R$, 1) < > "o" THEN VTAB 22: GOSUB
  2700
1720 IF O = 1 OR O$ = "L" OR O$ = "I" THEN
  1740
1730 GOSUB 2720
1740 PRINT D$; "PR#0": RUN
1750 REM
1760 FOR J = 1 TO N1
1770 K3 = J
1780 IF CA = 1 THEN 1810
1790 IF N(K3) = A3 THEN 1840
1800 GOTO 1820
1810 IF N(R(K3)) = A3 THEN 1840
1820 NEXT J
1830 PRINT : PRINT "PAS DE NUMERO
  D'ACTIVITE "; A3
1840 RETURN
1850 HOME
1860 PRINT CHR$(14); " "NE$: PRINT :
  PRINT : PRINT
1870 PRINT "CLASSEMENT ORDONNE
  DES EVENEMENTS : "
1880 PRINT "-----"
1890 PRINT : CE = 0
1900 FOR J = 1 TO N1: PRINT N(R(J)); "-> ";
1910 CE = CE + 1: IF CE > 13 THEN CE = 0:
  PRINT
1920 NEXT J: PRINT "FIN": PRINT : PRINT :
  PRINT
1930 IF LEFT$(R$, 1) < > "O" AND LEFT$(
  R$, 1) < > "o" THEN GOSUB 2700
1940 REM IMPRESSION DELAIS
1950 HOME : HTAB 12: INVERSE : PRINT "
  EVENEMENTS ": NORMAL : PRINT
1960 PRINT T$
1970 PRINT " — DELAI — "
1980 PRINT "EVENEMENT": HTAB 12:
  PRINT "AU + TOT": HTAB 22: PRINT "AU
  + TARD":
1990 HTAB 34: PRINT "ECART"
2000 PRINT T$
2010 FOR I = 1 TO N1
2020 PRINT N(R(I)): HTAB 15: PRINT INT
  (100 * E(I) + .5) / 100: HTAB 25: PRINT
  INT (100 * L(R(I)) + .5) / 100:
2030 HTAB 35: PRINT INT (100 * LAG(R(I))
  + .5) / 100
2040 NEXT I
2050 PRINT T$
2060 RETURN
2070 REM
2080 IF LEFT$(R$, 1) < > "O" AND LEFT$(

```

```

  (R$, 1) < > "o" THEN GOSUB 2700
2090 PRINT CHR$(12)
2100 T5 = 0: PRINT : PRINT
2110 HOME : HTAB 2: INVERSE : PRINT "
  ACTIVITES ET DUREES
  CORRESPONDANTES.": NORMAL :
  PRINT
2120 PRINT T$
2130 HTAB 22: PRINT "— TEMPS —"
2140 PRINT "ACTIVITES": HTAB 13: PRINT
  "DE": HTAB 19: PRINT "A": HTAB 21:
  PRINT "ATTENDU": HTAB 33: PRINT
  "MAX."
2150 PRINT T$
2160 CA = 1
2170 FOR I = 1 TO A
2180 A3 = P(I)
2190 GOSUB 1750
2200 I1 = K3
2210 A3 = S(I)
2220 GOSUB 1750
2230 I2 = K3
2240 D = L(R(I2)) - E(I1)
2250 DD(I) = D
2260 IF ABS(T(I) - D) < 0.001 THEN
  2300
2270 PRINT "A "; I; HTAB 13: PRINT P(I):
  HTAB 14: PRINT " -> "; PRINT S(I):
  HTAB 23: PRINT INT (100 * T(I) + .5) /
  100:
2280 HTAB 33: PRINT INT (100 * D + .5) /
  100
2290 GOTO 2330
2300 PRINT "A "; I; HTAB 13: PRINT P(I):
  HTAB 14: PRINT " -> "; PRINT S(I):
  HTAB 23: PRINT INT (100 * T(I) + .5) /
  100:
2310 HTAB 33: PRINT INT (100 * D + .5) /
  100: HTAB 5: INVERSE : PRINT
  "CRITIQUE": NORMAL
2320 T5 = T5 + T(I)
2330 NEXT I
2340 PRINT T$
2350 IF LEFT$(R$, 1) < > "O" AND LEFT$(
  R$, 1) < > "o" THEN GOSUB 2700
2360 HOME
2370 PRINT : PRINT : PRINT
2380 PRINT "VOICI L'ORDRE DU CHEMIN
  CRITIQUE : "
2390 PRINT "-----"
2400 FOR J = 1 TO N4: PRINT CP(J); "-> ";
2410 NEXT J
2420 PRINT "FIN"
2430 PRINT : PRINT
2440 PRINT "... AVEC UNE LONGUEUR
  ESTIMEE DE ": PRINT : HTAB 15:
  PRINT T5
2450 IF O = 1 THEN 2490
2460 HTAB 1: VTAB 15
2470 INPUT "NOM DU FICHIER A
  SAUVEGARDER ? "NE$
2480 IF NE$ = "" THEN 2460
2490 GOSUB 3130: REM SAUVEGARDE
  RESULTATS
2500 RETURN
2510 REM
2520 INPUT "LIBELLE DE L'OPERATION
  ": L$(I)
2530 INPUT "POINT DE DEPART ": P$: P(I)
  = VAL(P$)
2540 INPUT "POINT D'ARRIVEE ": S$: S(I)
  = VAL(S$)
2550 IF P(I) > = S(I) OR P(I) < = 0 THEN
  PRINT "< ERREUR >": GOTO 2530
2560 IF Q$ < > "C" AND Q$ < > "c" THEN
  2600
2570 INPUT "DUREE ": T$
2580 IF T$ < > "0" AND VAL(T$) = 0 THEN

```

```

  PRINT "< ERREUR >": GOTO 2570
2590 T(I) = VAL(T$): GOTO 2680
2600 INPUT "DELAI D'EXECUTION LE
  PLUS PROBABLE ": T1$
2610 IF T1$ < > "0" AND VAL(T1$) = 0
  THEN PRINT "< ERREUR >": GOTO 2600
2620 T1(I) = VAL(T1$)
2630 IF T1(I) = 0 THEN T2(I) = 0: T3(I) = 0:
  GOTO 2680
2640 INPUT " LE PLUS OPTIMISTE
  ": T2$: T2(I) = VAL(T2$)
2650 IF T2(I) > T1(I) OR (T2$ < > "0" AND
  VAL(T2$) = 0) THEN PRINT "< ERREUR
  >": GOTO 2640
2660 INPUT " LE PLUS
  PESSIMISTE ": T3$: T3(I) = VAL(T3$)
2670 IF T3(I) < T1(I) THEN PRINT "<
  ERREUR >": GOTO 2660
2680 RETURN
2690 REM
2700 PRINT : PRINT "APPUYEZ SUR UNE
  TOUCHE POUR LA SUITE.": GET C$
2710 RETURN
2720 REM
2730 PRINT
2740 PRINT D$; "OPEN": NE$
2750 PRINT D$; "WRITE": NE$
2760 PRINT A: PRINT O$
2770 FOR I = 1 TO A
2780 PRINT P(I): PRINT S(I): PRINT T(I):
  PRINT T1(I): PRINT T2(I): PRINT T3(I)
2790 PRINT L$(I)
2800 NEXT
2810 PRINT D$; "CLOSE"
2820 RETURN
2830 REM
2840 INVERSE : HTAB 20: VTAB 10: PRINT
  "UN PEU DE PATIENCE...": NORMAL
2850 PRINT
2860 PRINT D$; "OPEN": NE$
2870 PRINT D$; "READ": NE$
2880 INPUT A: INPUT O$
2890 A = A + 1: GOSUB 340: A = A - 1
2900 FOR I = 1 TO A
2910 INPUT P(I): INPUT S(I): INPUT T(I):
  INPUT T1(I): INPUT T2(I): INPUT T3(I)
2920 INPUT L$(I)
2930 NEXT
2940 PRINT D$; "CLOSE"
2950 RETURN
2960 REM
2970 IF PEEK(222) = 5 THEN VTAB 10:
  PRINT "FICHIER INEXISTANT": GOTO
  2990
2980 PRINT "ERREUR"
2990 FOR IT = 1 TO 1000: NEXT
3000 GOTO 1740
3010 REM
3020 HOME
3030 INPUT "NOM DU FICHIER A MODIFIER
  ? "NE$
3040 GOSUB 2850
3050 PRINT "VOULEZ - VOUS AJOUTER
  UNE ACTIVITE AUX "; A; " ACTIVITES
  EXISTANTES ";
3060 INPUT R$
3070 IF LEFT$(R$, 1) < > "O" AND LEFT$(
  R$, 1) < > "o" THEN
  3090
3080 A = A + 1: I = A: GOTO 3100
3090 INPUT "NUMERO DE L'ACTIVITE A
  MODIFIER ? "J
3100 GOSUB 2520
3110 GOSUB 2730
3120 RETURN
3130 REM
3140 PRINT D$; "OPEN": NE$; ".RESULT"
3150 PRINT D$; "WRITE": NE$; ".RESULT"

```

```

3160 PRINT A: PRINT N1
3170 FOR I = 1 TO N1
3180 PRINT N(R(I)): PRINT E(I): PRINT
      L(R(I))
3190 NEXT
3200 FOR I = 1 TO A
3210 PRINT P(I): PRINT S(I): PRINT T(I):
      PRINT DD(I)
3220 PRINT L$(I)
3230 NEXT
3240 PRINT N4
3250 FOR I = 1 TO N4
3260 PRINT CP(I)
3270 NEXT
3280 PRINT D$;"CLOSE"
3290 RETURN

```

Programme PERT.TEXT

```

20 HOME : HTAB 20: PRINT "LA METHODE
  P.E.R.T."
30 SPEED= 90
40 PRINT : PRINT "Cette méthode fut créée
  il y a quelques années pour
  l'établissement de"
50 PRINT "plannings de travaux de
  constructions.": PRINT "P.E.R.T. signifie
  Program Evaluation and Review
  Technique."
60 PRINT "Associée à la méthode du chemin
  critique,cette méthode est couramment
  employée"
70 PRINT "pour résoudre les problèmes
  d'ordonnement et d'organisation."
80 PRINT
90 PRINT "L'analyse du chemin critique
  permet ."
100 PRINT "- de detecter les taches
  critiques.,"
110 PRINT "c'est à dire celles dont
  l'exécution ne peut etre ni retardée,ni
  ralentie ";
120 PRINT "sans que le temps total des
  travaux ne soit": PRINT "reporté."
130 PRINT "- déterminer le meilleur temps
  total des travaux."
140 PRINT "- établir un ordonnancement."
150 PRINT : PRINT "Afin d'utiliser ce
  programme au mieux,il est conseillé de
  préparer un"
160 PRINT "tableau comprenant la liste des
  taches relatives aux travaux à effectuer"
170 PRINT "avec en face de chaque tache:
  le temps de réalisation le plus probable,"
180 PRINT "la durée
  minimale,"
190 PRINT "la durée
  maximale."
200 PRINT "Après constitution du tableau,il
  faudra dessiner sous forme de graphe,"
210 PRINT "l'ensemble des activités de
  maniere ordonnée."
220 PRINT "Ce graphe se composera de
  noeuds,c'est à dire des cercles
  numérotés"
230 PRINT "representant les
  événements.Chaque cercle sera relié à
  un autre cercle"
240 PRINT "par un arc,représentant les
  activités ou taches."
250 PRINT "Chaque noeud doit porter un
  numéro différent"
260 PRINT "le point d'arrivée d'un arc doit
  correspondre à un numéro de noeud
  supérieur"
270 PRINT "au noeud de départ."

```

```

280 PRINT "Le chemin defini est donc une
  suite continue d'arcs dont l'extrémité
  terminale"
290 PRINT "de l'un est l'extrémité initiale de
  l'autre.Il faut s'assurer que tous les"
300 PRINT "chemins du graphe ne sont pas
  interrompus.Si un chemin ne peut
  aboutir,"
310 PRINT "il est indispensable de créer un
  arc fictif de valeur nulle liant les"
320 PRINT "extrémités interrompues."
330 PRINT : PRINT "Le planning obtenu sur
  l'imprimante affichera.": PRINT "en haut
  les numéros de noeuds,"
340 PRINT "en ordonnée les activités,"
350 PRINT "en abscisse la durée."
360 PRINT : PRINT
370 SPEED= 255
380 PRINT "CE LOGICIEL A ETE INPIRE
  PAR L'ARTICLE DE XUAN TUNG BUI
  PARU"
390 PRINT "DANS MICRO-SYSTEME EN
  SEPTEMBRE-OCTOBRE 1982."
400 PRINT : INVERSE : INPUT "APPUYEZ
  SUR UNE TOUCHE POUR UNE
  DEMONSTRATION":RS$
410 NORMAL : PRINT
420 PRINT CHR$(4);"RUN PERT.DEMO"

```

Programme PERT.DEMO

```

40 & GR 10
50 & POS 110,180
60 & COLOR= 0: & BACK = 15
70 & PRINT "EXEMPLE.:"
80 & COLOR= 15: & BACK = 0
90 & PRINT "RESEAU DE
  CONSTRUCTION";
100 & POS 110,172: & PRINT "=====
  ";
110 & POS 1,136: & PRINT "1";
120 & PLOT 10,140 TO 40,140
130 & POS 25,144: & PRINT "a";
140 & POS 45,136: & PRINT "2";
150 & PLOT 60,140 TO 90,140
160 & POS 75,144: & PRINT "b";
170 & POS 95,136: & PRINT "3";
180 & PLOT 110,140 TO 140,140
190 & POS 125,144: & PRINT "c";
200 & POS 145,136: & PRINT "4";
210 & POS 175,143: & PRINT "e";
220 & PLOT 160,140 TO 190,140
230 & POS 195,136: & PRINT "5";
240 & POS 295,144: & PRINT "h";
250 & PLOT 280,140 TO 310,140
260 & POS 272,136: & PRINT "8";
270 & POS 345,144: & PRINT "i";
280 & PLOT 330,140 TO 360,140
290 & POS 315,136: & PRINT "9";
300 & PLOT 380,140 TO 410,140
310 & POS 395,144: & PRINT "k";
320 & POS 363,136: & PRINT "10";
330 & POS 413,136: & PRINT "11 Fin";
340 & PLOT 152,145 TO 230,162
350 & POS 167,155: & PRINT "d";
360 & POS 260,152: & PRINT "f";
370 & POS 235,165: & PRINT "6";
380 & PLOT 245,162 TO 270,145
390 & PLOT 152,135 TO 230,110
400 & POS 175,115: & PRINT "g";
410 & POS 235,100: & PRINT "7";
420 & PLOT 250,110 TO 365,135
430 & POS 285,110: & PRINT "j";
440 & PLOT 205,145 TO 238,160
450 & PLOT 205,135 TO 235,115
460 & POS 10,80: & PRINT "LES ARCS

```

```

ALLANT DU NOEUD 5 AU NOEUD 6,ET
DU NOEUD 5 AU NOEUD 7 SONT DES
ARCS";
470 & POS 10,70: & PRINT "FICTIFS DE
  VALEUR NULLE.:"
480 & COLOR= 0: & BACK = 15
480 & POS 10,30: & PRINT "APPUYEZ
  SUR UNE TOUCHE POUR LA LISTE
  DES TRAVAUX CORRESPONDANT A
  CE GRAPHE."
500 & COLOR= 15: & BACK = 0
510 GET A$
520 REM LISTE DE TRAVAUX
530 & TEXT 6
540 T$ = "=====
  ====="
550 INVERSE : POKE 36,28: PRINT
  "EXEMPLE DE LISTE DE TRAVAUX"
560 NORMAL : PRINT
570 PRINT T$
580 POKE 36,25: PRINT "TACHES";
590 POKE 36,47: PRINT "TEMPS DE
  REALISATION DES TRAVAUX"
600 POKE 36,47: PRINT "PROBABLE
  OPTIMISTE PESSIMISTE"
610 PRINT T$
620 PRINT "a l'Etablissement du plan";
630 POKE 36,50: PRINT "15";: POKE 36,62:
  PRINT "9";: POKE 36,73: PRINT "21"
640 PRINT "b l'Obtention du permis de
  construire";
650 POKE 36,50: PRINT "30";: POKE 36,62:
  PRINT "30";: POKE 36,73: PRINT "45"
660 PRINT "c l'Signature des contrats";
670 POKE 36,50: PRINT "5";: POKE 36,62:
  PRINT "3";: POKE 36,73: PRINT "10"
680 PRINT "d l'Transport du materiel au
  chantier";
690 POKE 36,50: PRINT "2";: POKE 36,62:
  PRINT "2";: POKE 36,73: PRINT "3"
700 PRINT "e l'Installation eau et
  electricite";
710 POKE 36,50: PRINT "3";: POKE 36,62:
  PRINT "3";: POKE 36,73: PRINT "3"
720 PRINT "f l'Fondations";
730 POKE 36,50: PRINT "10";: POKE 36,62:
  PRINT "8";: POKE 36,73: PRINT "15"
740 PRINT "g l'Transport de terre pour le
  jardin";
750 POKE 36,50: PRINT "2";: POKE 36,62:
  PRINT "2";: POKE 36,73: PRINT "13"
760 PRINT "h l'Murs";
770 POKE 36,50: PRINT "10";: POKE 36,62:
  PRINT "8";: POKE 36,73: PRINT "12"
780 PRINT "i l'Toit";
790 POKE 36,50: PRINT "12";: POKE 36,62:
  PRINT "7";: POKE 36,73: PRINT "14"
800 PRINT "j l'Decoration";
810 POKE 36,50: PRINT "10";: POKE 36,62:
  PRINT "8";: POKE 36,73: PRINT "15"
820 PRINT "k l'Decoration a l'interieur";
830 POKE 36,50: PRINT "10";: POKE 36,62:
  PRINT "9";: POKE 36,73: PRINT "13"
840 PRINT T$
850 INVERSE
860 VTAB 21: POKE 36,15: PRINT
  "APPUYEZ SUR LA TOUCHE 'Z' POUR
  RETOUR AU GRAPHE."
870 POKE 36,15
880 PRINT " SUR LA TOUCHE 'ENTER'
  POUR RETOUR AU MENU"
890 NORMAL
900 GET A$
910 IF A$ = "Z" THEN RUN
920 IF A$ < > CHR$(13) THEN 520
930 PRINT
940 PRINT CHR$(4);"RUN PERT"

```

Mémoire-tampon d'écran en Applesoft

Gérard Michel

Que vous soyez utilisateur d'un Apple II+ avec "carte langage", d'un Apple IIe ou d'un Apple IIc, vous ne disposez toujours que d'un système 48 K lorsque vous programmez en Applesoft, car les 16 K supplémentaires ne vous sont pas directement accessibles.

Pom's vous a déjà présenté plusieurs routines permettant d'exploiter cette réserve de mémoire : extensions du Basic s'y logeant en partie, comme Haifa, disque virtuel...

Nous vous proposons maintenant de l'utiliser comme zone de stockage pour des écrans 40 colonnes et les variables qu'ils peuvent éventuellement contenir. Ceci permet de rappeler facilement des écrans nécessaires en plusieurs points d'un programme, notamment ceux qui servent à la définition de paramètres pour votre application, mais aussi de faire passer ces écrans et variables d'un programme à un autre, sans qu'il soit besoin de les redéfinir dans le second.

Principe et mode d'emploi

Afin de préparer l'extension mémoire pour le stockage, il faut tout d'abord exécuter le petit programme d'initialisation listé plus loin par un BRUN INIT16K. Cela peut se faire dans le tout premier programme utilisant les routines concernées ou dans le programme de "boot".

Cette initialisation consiste simplement à mettre un 0 dans le premier octet de la zone de stockage, qui doit contenir à tout moment le nombre d'écrans effectivement stockés.

Pour ce qui est des routines de manipulation d'écrans, nous utiliserons, une fois encore, un accès via la modification de la routine \$B1, qui lit les octets du programme pour le compte de l'interpréteur Applesoft.

Votre premier programme devra donc commencer par les instructions de chargement suivantes (voir le listing de TESTVID16 donné plus bas) :

- BLOAD VID162 : chargement (à l'adresse \$9000) de la routine qui assure le traitement des instructions de manipulation d'écrans.
- BLOAD INTER : mise en place dans la routine \$B1 de l'aiguillage qui conduit en \$9000 sur VID162.

Dès lors, votre Basic dispose de 4 instructions nouvelles dédiées à notre mémoire tampon d'écran. C'est le caractère "]" qui identifie ces instructions, qu'il doit toujours précéder ; lui-

même doit toujours être précédé de ":", même s'il vient en tout début de ligne ou après un THEN.

- :JS

Stocke l'écran actuellement affiché dans l'extension mémoire, sans modifier l'affichage. Vous pouvez "ranger" ainsi 12 écrans ; au-delà, vous obtiendrez un message "OUT OF MEMORY".

Chaque écran archivé est implicitement numéroté dans l'ordre chronologique de son archivage. Le premier porte logiquement le numéro 1, et ainsi de suite. Il n'est donc pas nécessaire (et même pas permis...) de préciser ici le numéro d'ordre de l'écran stocké.

- :JR,I,Z\$

Retrouve dans l'extension mémoire l'écran numéro I, l'affiche et transfère dans le tableau Z\$ les variables qu'il peut contenir.

Vous pouvez spécifier le numéro d'écran par tout moyen à votre convenance : chiffres, variable, expression numérique...

Le tableau récepteur des données doit, de préférence, être dimensionné avant l'appel de fonction, sauf si la dimension par défaut peut lui convenir. Il doit obligatoirement appartenir au type alphanumérique et ne comporter qu'une seule dimension. Pour la routine de restitution d'écran, toute partie de cet écran comprise entre deux caractères ":" constitue une chaîne de caractères qui doit être affectée comme contenu pour l'un des éléments du tableau.

Si votre écran I comporte à un endroit donné les caractères suivants :

:DONNEE 1:

:DONNEE 2:

:DONNEE 3:

et que vous utilisez l'instruction

:JR,I,T\$

vous aurez pour résultat, outre le rappel à l'affichage de l'écran I :

T\$(0)=DONNEE 1

T\$(1)=DONNEE 2

T\$(2)=DONNEE 3

Pour la dernière donnée de l'écran, toutefois, si vous manquez de place en "VTAB 24 : HTAB 40" pour placer un ":", le ":" initial suffira pour réaliser l'affectation de la portion d'écran qui le suit au dernier élément du tableau.

Si vous spécifiez un numéro d'écran I qui n'existe pas dans l'extension mémoire, vous obtiendrez un message "ILLEGAL QUANTITY".

- :JD,I

Détruit l'écran numéro I dans l'exten-

sion mémoire. Les écrans qui le suivent, s'il y en a, sont décalés vers le bas de la mémoire et leur numérotation implicite est remise à jour en conséquence.

Ainsi, si vous détruisez l'écran numéro 3, le numéro 4 devient le numéro 3, le 5 devient le 4...

Chaque fois que vous supprimez un écran de cette façon, vous libérez de la place pour en stocker un autre, mais il ne portera évidemment pas le même numéro. Par exemple, si vous avez dix écrans en mémoire et que vous donnez les instructions suivantes :

:JD,1:JS

le nouvel écran archivé par JS portera le numéro implicite 10, et non 1 (voir ci-après pour la modification du contenu d'un écran archivé sous un numéro donné).

Là encore, si vous spécifiez un numéro I inexistant, vous obtiendrez un "ILLEGAL QUANTITY" en retour.

- :JU,I

Stocke l'écran actuellement affiché à l'emplacement réservé à l'écran numéro I dans l'extension mémoire.

Il peut s'agir du même écran préalablement rappelé pour modification (une nouvelle saisie de la valeur des variables, par exemple) ou d'un écran totalement différent, qui écrasera donc l'ancien porteur du numéro I.

La valeur de I doit correspondre à un écran effectivement archivé, sous peine de "ILLEGAL QUANTITY".

Précisions techniques et bibliographiques

Vous trouverez dans le courrier des lecteurs du Pom's 10 une explication complète sur le mode d'utilisation de l'extension mémoire 16 K.

Nous utilisons ici le Bank 2 (4 K) et la partie commune de 8 K, soit un ensemble de 12 K, dont les adresses, de \$D000 à \$FFFF, sont les mêmes que celles de la ROM de l'Apple. Un ensemble d'adresses joue le rôle de commutateurs afin de déterminer qui, de la ROM ou de l'extension RAM, est sélectionné en lecture ou écriture.

Les commutateurs exploités dans notre routine sont :

- \$C080 : sélectionne la RAM en lecture mais interdit l'écriture.
- \$C081 : sélectionne la ROM en lecture et autorise l'écriture sur la RAM si adressé deux fois (par

exemple, avec BIT \$C081: BIT \$C081, dans nos routines).

- \$C083: sélectionne la RAM en lecture et écriture si adressé deux fois.

Dans l'extension RAM, le premier octet (à l'adresse \$D000) contient le nombre d'écrans stockés, ces derniers s'empilent ensuite à partir de \$D001. Chacun d'entre eux occupe \$3C0 octets (soit 24 x 40 caractères = 960 octets).

En ce qui concerne la manipulation de la routine \$B1, pour implantation de nouvelles instructions sous Applesoft, vous pouvez notamment vous reporter au Pom's 16, dans lequel ces mécanismes sont présentés en détail à l'occasion de l'analyse du RUN.

Le petit fichier CHRGET listé ci-après doit être chargé en mémoire

par BLOAD si vous désirez retrouver l'Applesoft standard et supprimer l'indirection menant en \$9000.

Pour ce qui est de la structure des tableaux et du mode de stockage des éléments, vous trouverez tous les renseignements complémentaires, s'ils vous sont nécessaires, dans des numéros précédents de Pom's (12 et 17, par exemple).

Les quelques routines empruntées à la ROM de l'Apple dans le cadre de notre programme sont:

- \$FBC1: calcule l'adresse base de la ligne dont le numéro est indiqué dans \$25 (position verticale du curseur) et stocké dans l'accumulateur. Cette adresse est retournée aux adresses \$28 - \$29, qui peuvent donc servir de base pour un adressage indirect indexé par Y.

- \$DEBE: vérifie la présence d'une virgule à l'adresse pointée par le pointeur de programme Applesoft (\$B8 - \$B9) et lit le caractère suivant.
- \$DD67: évalue l'expression numérique ou algébrique pointée par \$B8 - \$B9 et range le résultat en flottant aux adresses \$9D à \$A2.
- \$E752: convertit le résultat ci-dessus en entier à deux octets et le range en \$50 - \$51.
- \$F7D9: recherche le tableau dont le nom est pointé par \$B8 - \$B9 et range l'adresse de son nom dans la zone de stockage des tableaux aux adresses \$9B et \$9C.
- \$D412: génère les messages d'erreur ou branche sur la routine ONERR GOTO si ce dernier est actif.

Programme INIT16.S (Assembleur BigMac)

```
1      ORG  $9000
2      BIT  $C081
3      BIT  $C081
4      LDA  #0
5      STA  $D000
6      BIT  $C081
7      CLC
8      JMP  $D88A
```

Programme VID162.REM.S (Assembleur BigMac)

```
1 *****
2 *
3 * STOCKAGE D'ECRANS ET VARIABLES*
4 * DANS LA MEMOIRE AUXILIAIRE //E*
5 * OU LA CARTE-LANGAGE II+ (16K) *
6 *      CODE = VID162      *
7 *
8 *****
9
10     ORG  $9000
11     CMP  #$5D      ;CARACTERE "]" ?
12     BEQ  S0
13     CMP  #$3A      ;REPRISE CHRGET
14     BCC  S1
15     RTS
16     S1   JMP  $BE
17     S0   JSR  $B1      ;LIT CAR SUIVNT ]
18     CMP  #'S'      ;CDE "SAUVE" ?
19     BEQ  S2
20     CMP  #'R'      ;CDE "RECUPERE" ?
21     BNE  D0
22     JMP  R0
23     D0   CMP  #'D'      ;CDE "DETUIT" ?
24     BNE  U0
```

```
25     JMP  D1
26     U0   CMP  #'U'      ;CDE "UPDATE" ?
27     BNE  E0      ;NON => ERREUR
                        (RTS DIRECT LA
                        PROVOQUERA)

28     JMP  D1
29     S2   STA  $19      ;SAUVE LA CDE
30     BIT  $C080      ;LECTURE MEV AUX.
31     LDA  $D000      ;NOMBRE D'ECRANS
                        ACTUELLEMENT
                        STOCKES
32     CMP  #$0C      ;12 ECRANS MAXI
33     BCC  S3
34     LDX  #$4D      ;"OUT OF MEMORY"
35     BIT  $C081      ;LECTURE ROM
36     JMP  $D412      ;ROUTINE D'ERREUR
                        DE L'APPLESOFT

37     S3   TAX
38     LDA  #1
39     STA  $6      ;INITIALISE $6-$7
40     LDA  #$D0
41     STA  $7
42     S5   CPX  #0
43     BEQ  S4
44     S50  CLC
45     LDA  $6      ;AJOUTE $3C0 PAR
                        ECRAN STOCKE
46     ADC  #$C0
47     STA  $6
48     LDA  $7
49     ADC  #3
50     STA  $7
51     DEX
52     BNE  S50
53     S4   LDA  #0      ;$6-$7 = ADRESSE
                        DE STOCKAGE DE
                        L'ECRAN
54     STA  $25      ;= NO LIGNE ECRAN
55     S9   BIT  $C081      ;LECTURE ROM
56     BIT  $C081      ;PERMET ECRITURE
                        SUR MEV
```

57	JSR	\$FBC1	;CALC ADRESSE BASE LIGNE	103	BEQ	R3	ETRE POSITIF
58	LDY	#0		104	STA	\$18	
59 S6	LDA	(\$28),Y	;LIT CARACTERE DANS LA LIGNE	105	JSR	\$DEBE	
60	STA	(\$6),Y	;ET LE STOCKE EN MEV AUX.	106	JSR	\$F7D9	;CHERCHE TABLEAU POINTE PAR B8-B9
61	INY			107	LDA	\$9B	
62	CPY	#\$28	;FIN DE LIGNE ?	108	STA	\$8	
63	BNE	\$6		109	LDA	\$9C	
64	INC	\$25	;LIGNE SUIVANTE	110	STA	\$9	; \$8-\$9 = ADR DU NOM DU TABLEAU
65	LDA	\$25		111	LDY	#4	
66	CMP	#\$18	;DERNIERE LIGNE DEJA TRAITEE ?	112	LDA	(\$8),Y	
67	BEQ	\$7		113	CMP	#1	;DOIT ETRE A UNE SEULE DIMENSION
68	CLC			114	BNE	R3	
69	LDA	\$6		115	CLC		
70	ADC	#\$28		116	LDA	#7	
71	STA	\$6	;PASSE A LA "LIGNE" SUIVANTE EN MEV AUX.	117	ADC	\$8	
72	BCC	\$8		118	STA	\$8	
73	INC	\$7		119	BCC	R4	
74 S8	LDA	\$25		120	INC	\$9	
75	BNE	\$9		121 R4	LDX	\$18	; \$8-\$9 = ADR 1ER ELEMENT
76 S7	DEC	\$25	;REVIENT LIGNE 24	122	DEX		
77	LDA	\$19		123	LDA	#1	
78	CMP	#'U'		124	STA	\$6	
79	BNE	\$70		125	LDA	#\$D0	
80	JMP	R140	;RETOUR SI UPDATE	126	STA	\$7	
81 S70	BIT	\$C083	;AUTORISE LECTURE	127 R6	CPX	#0	
82	BIT	\$C083	;ECRIT -> MEV AUX	128	BEQ	R5	
83	INC	\$D000	;INC NB ECRAN	129	CLC		
84	BIT	\$C081	;ROM EN LECTURE	130	LDA	\$6	
85	JMP	\$B1	;RETOUR APPLESOFT	131	ADC	#\$C0	
86 R0	JSR	\$B1	;POINTE \$B8-\$B9 SUR CARACTERE SUIVANT]	132	STA	\$6	
87	JSR	\$DEBE	;VERIFIE ", " ET POINTE SUR OCTET SUIVANT	133	LDA	\$7	
88	JSR	\$DD67	;EVALUE FORMULE POINTEE PAR \$B8-\$B9	134	ADC	#3	
89	JSR	\$E752	;LA CONVERTIT EN 2 OCTETS ET LES RANGE EN \$50-\$51	135	STA	\$7	
90	LDA	\$51		136	DEX		
91	BEQ	R1	;POIDS FORT DOIT ETRE A 0	137	JMP	R6	
92 R3	LDX	#\$35	; "ILLEGAL QUANT. "	138 R5	LDA	#0	; \$6-\$7=ADR ECRAN EN MEV AUX.
93	JMP	\$D412		139	STX	\$1C	
94 R1	BIT	\$C080	;LECTURE MEV AUX.	140	STA	\$19	;DRAPEAU POUR ":" ENTRE VARIABLES
95	LDA	\$D000		141	TAX		
96	CMP	\$50		142	STA	\$25	
97	BCS	R2	;NO ECRAN <= NOMBRE EN STOCK	143 R12	JSR	\$FBC1	
98	BIT	\$C081	;RETOUR ROM	144	BIT	\$C080	;LECTURE MEV AUX.
99	JMP	R3		145	LDY	#0	
100 R2	BIT	\$C081		146 R9	LDA	(\$6),Y	;REMET LES OCTETS A L'ECRAN
101	LDA	\$50		147	STA	(\$28),Y	
102	CMP	#0	;NO ECRAN DOIT	148	CMP	#\$BA	; ":" ?
				149	BEQ	R16	
				150	JMP	R15	
				151 R16	LDA	\$19	
				152	BEQ	R7	;1ER DELIMITEUR
				153	JSR	XX0	;C'EST LE 2ND => AFFECTER LA VARIABLE
				154	JMP	X1	

155	XX0	.STY \$1A	;SAUVEGARDE INDEX	195	LDY \$1A	
156		CPX #0	;LONG CHAINE=0 ?	196	LDX #0	
157		BNE X0		197	BEQ R8	
158		LDY #0		198	R7 INC \$19	;SIGNAL : 1ER
159		TXA				DELIMITEUR EST
160		STA (\$8),Y	;STOCKER SIMPLE-			TROUVE
			MENT LA LONGUEUR	199	R8 INY	
161		RTS	;CONCERNE DU	200	CPY #\$28	;POUR SUIT
			TABLEAU			AFFICHAGE ECRAN
162	X0	STX \$1B	;SAUVEGARDE LONG	201	BNE R9	
			CHAINE	202	INC \$25	
163		SEC		203	LDA \$25	
164		LDA \$6F	;CALC ADR CHAINE	204	CMP #\$18	
			EN RETIRANT	205	BEQ R10	
165		SBC \$1B	;SA LONGUEUR DE	206	CLC	
			L'ADRESSE	207	LDA \$6	
			POINTEE PAR	208	ADC #\$28	; "LIGNE" SUIVANTE
166		STA \$6F	;FRETOP (\$6F-\$70)			EN MEV AUX.
			= HAUT MEMOIRE	209	STA \$6	
167		BCS X2		210	BCC R11	
168		DEC \$70		211	INC \$7	
169	X2	LDX #0		212	R11 BIT \$C081	;REMET LECTURE
170		LDY #0				ROM POUR LE
171	X4	LDA BUFF,X	;TRANSFERE LE			"JSR \$FBC1"
			CONTENU DE LA	213	LDA \$25	
			CHAINE A SON	214	JMP R12	
172		STA (\$6F),Y	;ADRESSE DANS LA	215	R10 BIT \$C081	
			ZONE DE STOCKAGE	216	CPX #0	
			DES CHAINES	217	BNE R13	
173		INX		218	R14 DEC \$25	
174		CPX \$1B	;CHAINE	219	LDX \$1C	
			INTEGRALEMENT	220	R140 JSR \$B7	;RETOUR A
			TRANSFERE ?			L'APPLESOFT
175		BEQ X3				(\$B8-\$B9 POINTE
176		INY				DEJA SUR
177		BNE X4		221	RTS	;L'OCTET QUI SUIV
178	X3	LDY #0				LE NOM DU
179		LDA \$1B	;MET LONG CHAINE			TABLEAU)
			DANS L'ELEMENT	222	R15 STA \$1D	
180		STA (\$8),Y	;DU TABLEAU	223	LDA \$19	
			CONCERNE	224	BEQ R8	
181		INY		225	LDA \$1D	
182		LDA \$6F	;AINSI QUE	226	STA BUFF,X	;STOCKE BUFFER DE
			L'ADRESSE DE LA			VARIABLES :
			CHAINE	227	INX	;ON EST APRES LE
			CORRESPONDANTE			1ER DELIMITEUR
183		STA (\$8),Y		228	BNE R8	
184		INY		229	R13 JSR XX0	;STOCKE UNE EVEN-
185		LDA \$70				TUELLE DERNIERE
186		STA (\$8),Y				VARIABLE DONT
187		RTS		230	JMP R14	;LE DELIMITEUR DE
188	X1	CLC				FIN NE SERAIT
189		LDA \$8	;POINTE SUR			PAS A L'ECRAN
			L'ELEMENT DU	231	D1 STA \$19	;STOCKE LA
			TABLEAU SUIVANT			COMMANDE
190		ADC #3		232	JSR \$B1	
191		STA \$8		233	JSR \$DEBE	
192		BCC X5		234	JSR \$DD67	
193		INC \$9		235	JSR \$E752	
194	X5	DEC \$19	;REMET LE DRAPEAU	236	LDA \$51	
			DELIMITEUR A 0	237	BEQ D2	
				238	JMP R3	

239 D2	BIT	\$C080		293	DEX	
240	LDA	\$D000		294	BNE	D70
241	BIT	\$C081		295 D8	CLC	; \$8-9 = ADR ECRAN A DETRUIRE
242	TAX					
243	CMP	\$50		296	LDA	\$8
244	BCS	D3		297	ADC	# \$C0
245	JMP	R3		298	STA	\$1A ; \$1A-1B=ADR ECRAN SUIVANT
246 D3	LDA	\$50	; \$50 CONTIENT UN NUMERO D'ECRAN VALIDE	299	STA	\$1C ; \$1C-\$1D SERVIRA POUR LE DECALAGE
				300	LDA	\$9
247	BNE	D4	; SAUF SI 0	301	ADC	#3
248	JMP	R3		302	STA	\$1B
249 D4	STA	\$18	; GARDE NO ECRAN	303	STA	\$1D
250	LDA	\$19		304	BIT	\$C083 ; MEV AUX. EN LECT/ECRITURE
251	CMP	# 'U'				
252	BNE	DD4		305	BIT	\$C083
253	DEC	\$18	; DEC NO POUR UTILISER LA MEME	306	LDY	#0
				307 D9	LDA	(\$1C), Y ; RECOPIE ECRAN SUIVANT A L'A- DRESSE DE CELUI
254	LDA	\$18	; PROCEDURE QUE POUR LA CDE "S"			
255	JMP	S3		308	STA	(\$8), Y ; QUI LE PRECEDE
256 DD4	CPX	#1	; RIEN A FAIRE SI UN SEUL ECRAN	309	INC	\$8
				310	BNE	D10
257	BNE	D40		311	INC	\$9
258	BIT	\$C083	; SINON ALLER	312 D10	LDA	\$9
DIMINUER LE NOMBRE D'ECRANS STOCKES				313	CMP	\$1B ; DECALAGE ECRAN COMPLET FINI ?
259	BIT	\$C083	; ET RETOUR APPLESOFT			
				314	BNE	D11
260	JMP	D99		315	LDA	\$8
261 D40	DEX			316	CMP	\$1A
262	LDA	#1		317	BEQ	D12
263	STA	\$6		318 D11	INC	\$1C ; NON => CONTINUE
264	LDA	# \$D0		319	BNE	D9
265	STA	\$7		320	INC	\$1D
266 D5	CPX	#0		321	BNE	D9
267	BEQ	D6		322 D12	LDA	\$1B ; DERNIER ECRAN STOCKE EST-IL MAINTENANT ; DECALE ?
268 D50	CLC					
269	LDA	\$6		323	CMP	\$7
270	ADC	# \$C0		324	BNE	D130
271	STA	\$6		325	LDA	\$1A
272	LDA	\$7		326	CMP	\$6
273	ADC	#3		327	BEQ	D99
274	STA	\$7		328 D130	INC	\$1C ; NON => ECRAN SUIVANT
275	DEX			329	BNE	D13
276	BNE	D50		330	INC	\$1D
277 D6	LDA	\$18	; \$6-\$7 = ADR DU DERNIER ECRAN	331 D13	LDA	\$1C
				332	STA	\$1A
278	TAX			333	LDA	\$1D
279	DEX			334	STA	\$1B
280	LDA	#1		335	BNE	D9
281	STA	\$8		336 D99	DEC	\$D000 ; DEC NBR ECRANS STOCKE
282	LDA	# \$D0				
283	STA	\$9		337	BIT	\$C081 ; MET ROM EN LECT ; RETOUR APPLESOFT
284 D7	CPX	#0		338	JSR	\$B7
285	BEQ	D8		339	RTS	
286 D70	CLC			340 BUFF	EQU	\$200 ; BUFFER D'ENTREE UTILISE ICI POUR
287	LDA	\$8				
288	ADC	# \$C0		341 *		; STOCKER TEMPORAIREMENT LES VARIABLES
289	STA	\$8				
290	LDA	\$9				
291	ADC	#3				
292	STA	\$9				

Programme TESTVID16

```
10 HIMEM: 9 * 4096 - 1
20 D$ = CHR$(4): PRINT D$ "BLOAD VID162": PRINT
  D$ "BLOAD INTER"
30 TEXT : HOME : DIM V$(10)
40 GOTO 100
50 FOR I = 0 TO 10: PRINT "LIGNE NO "I": DONNEE NO
  "I;J": PRINT : NEXT : RETURN
100 FOR J = 1 TO 12: TEXT : HOME : GOSUB 50: GET
  Z$:J$ : NEXT
105 TEXT : HOME : GET Z$
110 FOR I = 1 TO 12:J,R,I,V$: GET Z$: TEXT : HOME : FOR
  J = 0 TO 10: PRINT V$(J): NEXT : GET Z$: NEXT
120 FOR I = 2 TO 12 STEP 2:J,R,I,V$: VTAB 24: HTAB 1:
  PRINT "ECRAN "I" DEVIENT ECRAN "I - I/2": GET
  Z$:J,U,I: NEXT
130 FOR I = 1 TO 6:J,D,I: NEXT
140 PRINT : PRINT D$ "RUN TEST2VID16"
```

Programme TEST2VID16

```
10 HIMEM: 9 * 4096 - 1
30 TEXT : HOME : DIM V$(10)
```

Indirection CHRGET

00BA- C9 3A B0

Indirection INTER

00BA- 4C 00 90

Récapitulation INIT16K

9000- 2C 81 C0 2C 81 C0 A9 00
9008- 8D 00 D0 2C 81 C0 18 4C
9010- 8A D8 FF

Récapitulation VID162

9000- C9 5D F0 08 C9 3A 90 01
9008- 60 4C BE 00 20 B1 00 C9
9010- 53 F0 15 C9 52 D0 03 4C
9018- A2 90 C9 44 D0 03 4C CC
9020- 91 C9 55 D0 E3 4C CC 91
9028- 85 19 2C 80 C0 AD 00 D0
9030- C9 0C 90 08 A2 4D 2C 81
9038- C0 4C 12 D4 AA A9 01 85
9040- 06 A9 D0 85 07 E0 00 F0
9048- 10 18 A5 06 69 C0 85 06
9050- A5 07 69 03 85 07 CA D0
9058- F0 A9 00 85 25 2C 81 C0
9060- 2C 81 C0 20 C1 FB A0 00
9068- B1 28 91 06 C8 C0 28 D0

9070- F7 E6 25 A5 25 C9 18 F0
9078- 0F 18 A5 06 69 28 85 06
9080- 90 02 E6 07 A5 25 D0 D5
9088- C6 25 A5 19 C9 55 D0 03
9090- 4C B4 91 2C 83 C0 2C 83
9098- C0 EE 00 D0 2C 81 C0 4C
90A0- B1 00 20 B1 00 20 BE DE
90A8- 20 67 DD 20 52 E7 A5 51
90B0- F0 05 A2 35 4C 12 D4 2C
90B8- 80 C0 AD 00 D0 C5 50 B0
90C0- 06 2C 81 C0 4C B2 90 2C
90C8- 81 C0 A5 50 C9 00 F0 E2
90D0- 85 18 20 BE DE 20 D9 F7
90D8- A5 9B 85 08 A5 9C 85 09
90E0- A0 04 B1 08 C9 01 D0 CA
90E8- 18 A9 07 65 08 85 08 90
90F0- 02 E6 09 A6 18 CA A9 01
90F8- 85 06 A9 D0 85 07 E0 00
9100- F0 11 18 A5 06 69 C0 85
9108- 06 A5 07 69 03 85 07 CA
9110- 4C FE 90 A9 00 86 1C 85
9118- 19 AA 85 25 20 C1 FB 2C
9120- 80 C0 A0 00 B1 06 91 28
9128- C9 BA F0 03 4C B8 91 A5
9130- 19 F0 54 20 39 91 4C 74
9138- 91 84 1A E0 00 D0 06 A0
9140- 00 8A 91 08 60 86 1B 38
9148- A5 6F E5 1B 85 6F B0 02
9150- C6 70 A2 00 A0 00 BD 00
9158- 02 91 6F E8 E4 1B F0 03
9160- C8 D0 F3 A0 00 A5 1B 91
9168- 08 C8 A5 6F 91 08 C8 A5
9170- 70 91 08 60 18 A5 08 69
9178- 03 85 08 90 02 E6 09 C6
9180- 19 A4 1A A2 00 F0 02 E6
9188- 19 C8 C0 28 D0 96 E6 25

9190- A5 25 C9 18 F0 13 18 A5
9198- 06 69 28 85 06 90 02 E6
91A0- 07 2C 81 C0 A5 25 4C 1C
91A8- 91 2C 81 C0 E0 00 D0 16
91B0- C6 25 A6 1C 20 B7 00 60
91B8- 85 1D A5 19 F0 CB A5 1D
91C0- 9D 00 02 E8 D0 C3 20 39
91C8- 91 4C B0 91 85 19 20 B1
91D0- 00 20 BE DE 20 67 DD 20
91D8- 52 E7 A5 51 F0 03 4C B2
91E0- 90 2C 80 C0 AD 00 D0 2C
91E8- 81 C0 AA C5 50 B0 03 4C
91F0- B2 90 A5 50 D0 03 4C B2
91F8- 90 85 18 A5 19 C9 55 D0
9200- 07 C6 18 A5 18 4C 3C 90
9208- E0 01 D0 09 2C 83 C0 2C
9210- 83 C0 4C A5 92 CA A9 01
9218- 85 06 A9 D0 85 07 E0 00
9220- F0 10 18 A5 06 69 C0 85
9228- 06 A5 07 69 03 85 07 CA
9230- D0 F0 A5 18 AA CA A9 01
9238- 85 08 A9 D0 85 09 E0 00
9240- F0 10 18 A5 08 69 C0 85
9248- 08 A5 09 69 03 85 09 CA
9250- D0 F0 18 A5 08 69 C0 85
9258- 1A 85 1C A5 09 69 03 85
9260- 1B 85 1D 2C 83 C0 2C 83
9268- C0 A0 00 B1 1C 91 08 E6
9270- 08 D0 02 E6 09 A5 09 C5
9278- 1B D0 06 A5 08 C5 1A F0
9280- 08 E6 1C D0 E6 E6 1D D0
9288- E2 A5 1B C5 07 D0 06 A5
9290- 1A C5 06 F0 10 E6 1C D0
9298- 02 E6 1D A5 1C 85 1A A5
92A0- 1D 85 1B D0 C6 CE 00 D0
92A8- 2C 81 C0 20 B7 00 60 00

```
40 FOR I = 1 TO 6:J,R,I,V$: GET Z$: TEXT : HOME : FOR J
  = 0 TO 10: PRINT V$(J): NEXT : GET Z$: NEXT
50 DIM W$(11): TEXT : HOME : FOR I = 0 TO 11: PRINT
  "VARIABLE NO "I": INPUT W$(I): NEXT
60 :J,R,6,V$: FOR I = 0 TO 11: VTAB 2 * I + 1: HTAB 14:
  PRINT W$(I): CALL - 868: NEXT : VTAB 23: HTAB
  1: PRINT "NOUVEAU ": GET Z$
70 :J$: PRINT : PRINT CHR$(4) "RUN TEST3VID16"
```

Programme TEST3VID16

```
10 TEXT : HOME : DIM V$(11)
20 FOR I = 1 TO 7:J,R,I,V$: GET Z$: TEXT : HOME : FOR J
  = 0 TO 10: PRINT V$(J): NEXT : GET Z$: NEXT
30 :J,R,1,V$: VTAB 23: HTAB 1: PRINT "CET ECRAN VA
  ETRE RECOPIE SUR": PRINT "LES 6 AUTRES...":
40 FOR I = 1 TO 7:J,U,I: NEXT : PRINT : PRINT CHR$(
  4) "RUN TEST4VID16"
```

Programme TEST4VID16

```
10 TEXT : HOME : DIM V$(11)
20 FOR I = 1 TO 7:J,R,I,V$: GET Z$: TEXT : HOME : FOR J
  = 0 TO 10: PRINT V$(J): NEXT : GET Z$: NEXT
```

Nous allons parler, ici, d'intelligence artificielle. La première chose à faire, puisque nous nous supposons intelligents, est de définir ce que "parler" et "intelligence" veulent dire.

"Parler" ici signifie "communiquer une information". On transmet une information à travers un système de communication. Un système de communication est composé de trois éléments : un émetteur, le moyen utilisé et un récepteur. Si pour une raison quelconque le récepteur reçoit l'information déformée, on peut dire qu'il n'y a pas eu communication, donc qu'on n'a pas "parlé". Par exemple, si vous utilisez la parole comme moyen et si vous parlez au-vergnat à un chinois, il ne "recevra" pas, donc vous avez émis des sons, mais vous n'avez pas "parlé" (d'après la définition précédente). La plupart des problèmes d'incompréhension (information déformée) viennent de l'utilisation des mots abstraits. Ceux-ci sont des commodités du langage qui réunissent un ensemble d'actions ou de qualités. Rapidité "contient" une caractéristique de déplacement, blancheur réunit une des caractéristiques des objets blancs. On voit donc, qu'au delà du mot abstrait, il y a quelque chose de tout à fait réel, de physique et de bien définissable.

Dans tous discours, autres que des propos de bistrot, et particulièrement dans des conversations scientifiques, les mots abstraits doivent être définis au préalable (vous allez rire, mais il est parfois bien utile de définir même les mots non abstraits).

Ce préambule n'est sûrement pas inutile, car nous abordons quelque chose de plus complexe : définir des mots abstraits, dont tout le monde est convaincu de bien connaître la signification.

Commençons par "Intelligence", nous verrons plus loin le mot "Amour".

L'évolution simulée

Dans le livre de Fogel, Owens et Walsh (voir bibliographie), on trouve une méthode pour simuler sur un ordinateur, ce que nous appelons l'intelligence. Leur premier soin a été de définir l'intelligence, c'est la moindre des choses si on veut savoir comment faire fonctionner le modèle. "L'intelligence est la capacité de toute entité capable de décision, d'atteindre un succès total ou partiel en poursuivant une grande quantité de buts dans des circonstances très différentes". Nous voyons déjà qu'il faut faire une distinction entre la connaissance et l'intelligence. Celui dont la culture et les connaissances sont étendues n'est pas automatiquement intelligent; par contre, celui qui sait

L'intelligence Artificielle

Guido Bettiol

utiliser son savoir pour atteindre un but l'est.

Fogel et ses amis ont voulu simuler un comportement intelligent sur une machine et ont défini son intelligence, donc l'intelligence artificielle, comme la manière d'atteindre un but par un moyen auquel l'homme n'aurait pas pensé. Ils ont pris comme point de départ le phénomène de l'évolution de la vie sur terre et ils ont rentré ce schéma de fonctionnement dans l'ordinateur. Ils ont établi un modèle qui est la représentation mathématique de l'organisme à étudier. Le comportement de ce modèle simule une caractéristique essentielle du comportement d'un homme dans une situation précise. Ce qui est important, est que ce modèle s'auto-modifie, donc évolue, pour obtenir

une sortie qui corresponde au but. La place manque ici pour en parler plus longuement, ces explications sont du type "vite fait sur le gaz" (voir leur livre). Avec cette approche on peut vraiment parler d'intelligence.

Les systèmes experts

Un système expert est un programme qui permet, à une grande variété de gens, d'avoir accès à des informations de haut niveau. Un système expert ne rendra pas intelligent un "crétin", ni ne donnera une formation à celui qui n'est pas un spécialiste, mais il permettra à quelqu'un qui possède déjà une certaine connaissance de la matière traitée, de résoudre des problèmes qui auraient

Pom's vous propose

"Dominos"

Thierry Haurie

Apple][+, //e, //c

Il est inutile de présenter le jeu de dominos; celui-ci bénéficie d'un graphisme très soigné (en couleur si vous disposez d'une carte "Chat Mauve") et les messages transmis par le programme sont, au choix, en Français, en Italien, en Allemand ou en Anglais.



80.00 F TTC franco
Bon de commande page 74

demandé la consultation d'un grand nombre de spécialistes. Il trouvera peut-être, même, une solution qu'un groupe de spécialistes n'aurait pas trouvée, car une seule entité (l'ordinateur) traite les données de tous les spécialistes.

Un système expert est un logiciel divisé en deux parties : la base de données et le moteur d'inférence.

Tout le monde sait ce qu'est une base de données, mais détruisons tout de suite un mythe : les bases de données ne sont pas intelligentes, elles sont tout à fait stupides, mais la masse de données qu'elles contiennent est précieuse. Reste à les utiliser "intelligemment". Le moteur d'inférence (d'après le Larousse, l'inférence est une "Opération intellectuelle par laquelle on passe d'une vérité à une autre vérité, jugée telle en raison de son lien avec la première") est donc un programme qui pose une question et reçoit une réponse. Il va alors chercher la réponse dans la base de données, en utilisant les règles qu'il possède.

Un système expert est-il intelligent ? En quelque sorte, oui. Un homme qui dispose des mêmes données appliquera théoriquement le même raisonnement, mais en pratique il utilisera des simplifications dues à ses propres connaissances et à son conditionnement. La force des machines est la stupidité bornée de la démarche systématique, comme la force de la police est la routine !

Que peut-on faire avec un Apple dans cette galère ? Compte tenu de la faible capacité mémoire, due à l'adressage limité du microprocesseur 8 bits, on ne peut pas aller très loin. La mémoire virtuelle augmente tout de même les possibilités.

Quelle que soit la complexité de leurs règles, une chose est commune aux systèmes experts : ils ne peuvent pas modifier eux-mêmes leurs règles en fonction de l'évolution des questions / réponses. Or la première nécessité d'un système qui se veut intelligent est de pouvoir modifier sa structure interne pour simuler le processus d'apprentissage.

Sur Apple on peut utiliser un système expert, commercialisé par MVP (voir la bibliographie), écrit en Forth. Il est fourni avec le source, est totalement transparent, peut être modifié (bon courage...) et enfin il est nettement plus rapide que le Basic.

Un exemple pratique

Les langages du type LISP sont bien adaptés aux problèmes d'intelligence artificielle; cependant, il faut programmer le moteur d'inférence.

Le langage Prolog a un énorme avantage : il EST un moteur d'infé-

rence. Nous parlerons, ici, de micro-Prolog, qui est anglais et assez proche du standard d'Edimbourg. Il est sous CP/M et nécessite une carte Z80.

Passons maintenant à un exercice périlleux en revenant au problème des mots abstraits. Nous allons bâtir un programme à travers lequel on définira le mot "Amour". Bien que tout le monde soit convaincu d'en connaître la signification, il est certain que la base de données "concrètes", formée de mots non abstraits que ce programme construira, étonnera plus d'un d'entre nous.

Le programme demande de définir l'amour. On peut répondre avec un mot abstrait ou avec un mot non abstrait. Si le mot est abstrait on en demande la définition, qui doit être un mot non abstrait. On demande ensuite d'associer ce dernier mot à une des deux "sources" possibles : SEX et BIO. Ainsi, on a construit deux tables : une avec un mot non abstrait associé à SEX ou BIO, l'autre avec un mot abstrait, qui à son tour est défini par un mot non abstrait, le tout associé à SEX ou BIO.

On pourra ensuite lister ces tables et les étudier. L'interprétation des résultats est encore un autre problème, et elle n'est pas envisagée ici.

On lance "question." et la réponse est affectée à la variable X.

Ensuite "tri X" est appelé qui à son tour appelle "abs X" (comme s'il s'agissait de sous-routines) lequel vérifie si le mot entré fait partie de l'ensemble de Faits "abstrait ..." (ici on a rentré seulement très peu de Faits, mais cet ensemble devrait en réalité contenir TOUS les mots ou expressions abstraites avec lesquels on risque de définir "amour").

Avec les résultats de ce dernier test, on revient en arrière. Si le mot entré était abstrait, on suit la branche "then", donc "définir-abs X", qui demande un mot non abstrait. On passe ensuite à "analogie X Y" pour l'association avec SEX ou BIO et on augmente la base de données d'un nouveau Fait. Ces Faits sont réunis par la relation "love", alors que les premières définitions sont réunies par la relation "amour".

Le listing 3 illustre le fonctionnement de ce programme. A noter que la première réponse, "a coté de ses pompes", bien que composée de mots non abstraits, est une expression abstraite et n'aurait pas dû être rentrée.

Le listing 4 est la liste des nouveaux Faits que l'on vient de créer. Ils sont réunis sous les relations "amour" et "love".

Voilà, notre ordinateur nous a permis de voir comment définir un mot abs-

trait sans équivoque et d'une manière reproductible. Ceci est certainement le premier pas à faire pour une approche de l'intelligence artificielle qui se propose d'explorer le cerveau humain et sa façon de raisonner. Le but, à long terme, est de reproduire ce fonctionnement.

Jusqu'où peut aller l'Apple avec micro-Prolog ? Le problème est toujours le même : la taille mémoire disponible et la vitesse. Cependant, on peut aller assez loin, puisqu'on peut utiliser la mémoire virtuelle pour stocker les données.

Bibliographie

Les langages

Système expert en Forth distribué par MVP (Mountain View Press), PO Box 4656, Mountain View, CA 94040, USA. Une disquette pour Apple et un manuel de 60 pages pour \$100.

PrologII est distribué par la Société Prologia, 278 rue St. Pierre, 13005 Marseille, tél. (91) 41 48 49. Trois disquettes et le manuel, 2965 Frs TTC.

micro-Prolog est distribué par LPA (Logic Programming Associated Ltd.), 10 Burntwood Close, London SW18 3JU, Grande Bretagne. Une disquette sous CP/M Z80 pour Apple, un manuel de 250 pages et le livre : micro-Prolog, Programming in Logic, Clark et McCabe, le tout pour \$195.

Les livres

Introduction aux systèmes experts, Michel Gondran, Eyrolles. Il donne une vue d'ensemble du sujet et possède une belle bibliographie.

Artificial intelligence through simulated evolution, Fogel, Owens et Walsh, John Wiley and Sons, 1966. Un livre passionnant de "haut niveau".

Artificial intelligence : an evolutionary idea, par Michael Wimble, parts 1 et 2, BYTE, mai et juin 1977. Organigramme d'une étude à partir du livre Artificial Intelligence... de Fogel et al, cité ci-dessus. Cet article est très intéressant. Il n'est sans doute pas inutile de citer la conclusion de Wimble : "D'après mon expérience personnelle : n'exposez pas cette technique à n'importe qui et n'importe quand. Beaucoup de gens ont peur des ordinateurs comme ils ont peur de tout ce qu'ils ne comprennent pas, et leur enthousiasme n'est peut-être pas égal au vôtre" (c'était en 77); les mentalités ont peut-être évoluées, mais ce n'est pas si sûr que ça !

Listing 1

Structuration de la recherche d'une définition de mot abstrait. On admet au départ que la notion "Amour" est liée en dernier lieu aux notions de "sexe" et "biologique", dont la définition est donnée. (Le travail a été effectué sur ThinkTank).

DEMANDER DEFINITION

Comparer le mot demandé ; la table des mots abstraits.

MOT NON-ABSTRAIT

Demander d'associer ce nom ; sex-bio.

MOT ABSTRAIT

Demander de définir le mot abstrait avec un mot non-abstrait.

Ensuite demander de l'associer ; sex-bio.

ASSOCIATION A SEX-BIO

Deux grandes classes: sex et bio.

Bio est incontrôlable.

Sex est contrôlable.

DEFINITIONS

BIO: Bio est le côté biologique du sexe.

Il "est", c'est tout, il est là. On se limite ; constater l'existence de la pulsion sexuelle.

SEX: Sex est le côté conscient du sexe. On sait qu'il existe et on le subit, mais on peut le contrôler, dans une certaine mesure (avec une motivation suffisante).

ON A 2 TABLES ASSOCIEES A SEX-BIO

Non-abstrait: (amour non-a)

Abstrait: (love abs (non-a sex-bio))

BASE DE DEFINITIONS

On dispose d'une base de données qui sont les définitions du mot "amour".

```
((analogue X Y)
  (mesure Z)
  (P Y "peut-il se rapprocher de " Z)
  (associer X Y Z))
((tri X)
  (IF (abs X) ((definir-abs X)) ((ADDCL ((amour
x))))))
((question .)
  (P "Ou'est-ce que l'amour? ")
  (R x)
  (tri X))
((définir-abs X)
  (PP X est un mot abstrait. Il faut le définir.)
  (P "Utilisez un mot non abstrait: ")
  (R Y)
  (analogue X Y))
((abstrait tendresse))
((abstrait extase))
((abstrait passion))
((abstrait bonheur))
```

LISTING 3

Dump écran du programme pendant son fonctionnement. (Voir texte).

question.

Ou'est-ce que l'amour? .extase

extase est un mot abstrait. Il faut le définir.

Utilisez un mot non abstrait: (à côté de ses pompes)

(à côté de ses pompes peut-il se rapprocher de sex (o/n) ? . o

question.

Ou'est-ce que l'amour? .passion

passion est un mot abstrait. Il faut le définir.

Utilisez un mot non abstrait: tachycardie

tachycardie peut-il se rapprocher de sex (o/n)? .n

tachycardie peut-il se rapprocher de bio (o/n)? .o

question.

Ou'est-ce que l'amour? .sexe

Listing 2

Programme écrit en micro-Prolog. La relation "abstrait" doit servir de référence pour détecter les mots ou expressions abstraites avec lesquelles on peut définir le mot "amour". La relation "mesure" matérialise les deux faits "sex" et "bio".

```
((abs X)
  (abstrait X)
  (EO X Y))
((associer X Y Z)
  (P " (o/n) ? ")
  (R x)
  (IF (EO x o) ((ADDCL ((love X (Y Z)))))
  ((analogue X Y))))
((mesure sex))
((mesure bio))
```

Listing 4

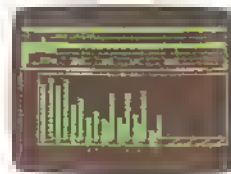
La trace du programme. On peut y suivre le travail effectué par le moteur d'inférence.

```
((love extase ((à côté de ses pompes) sex))
(love passion (tachycardie bio))
(amour sexe))
```

Puisque nous ne vous aviez besoin, nous



Apple et le logo Apple sont des marques déposées d'Apple Computer, Inc.



Version Calc

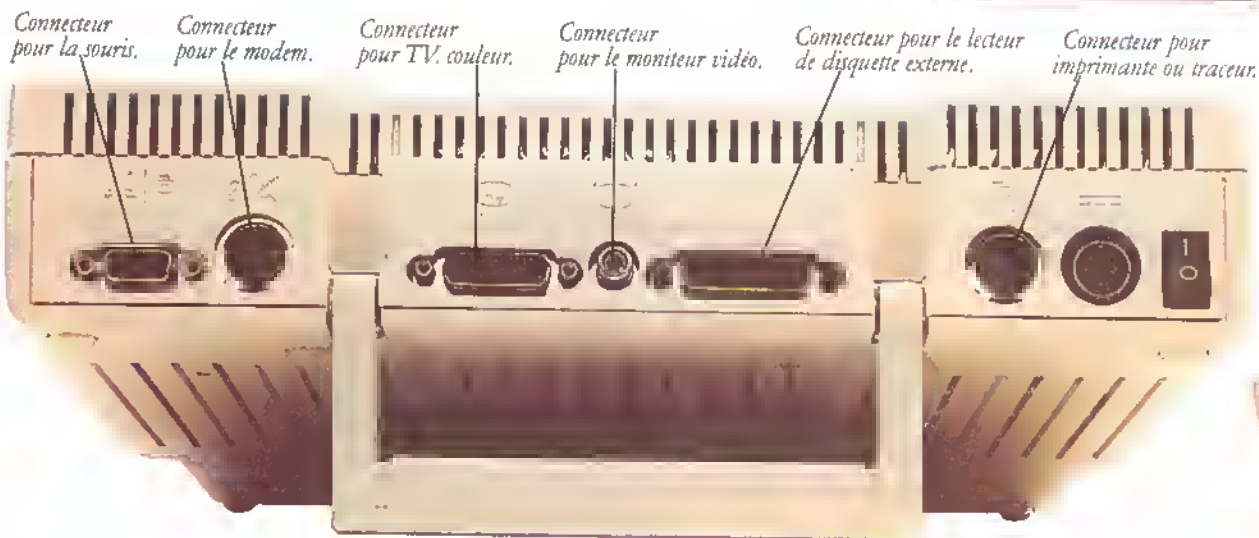


MousePaint

Une des plus grandes bibliothèques de logiciels programmes compatibles avec l'Apple IIe : jeux, gestion de base de données, analyse financière,



Un clavier 63 touches type AZERTY comprenant une accentuation complète et des caractères majuscules/minuscules intégrés.

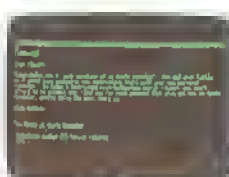


Voici comment l'écran de haute résolution se contrôle très facilement. Si nous vous en avions eu besoin, en un seul appareil était indispensable.

savons pas de quoi vous avons tout donné.



Budget Familial



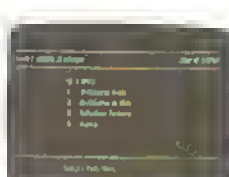
Apple Writer



Sorcellerie



Apple Works



Omni II



Les Oursins malins

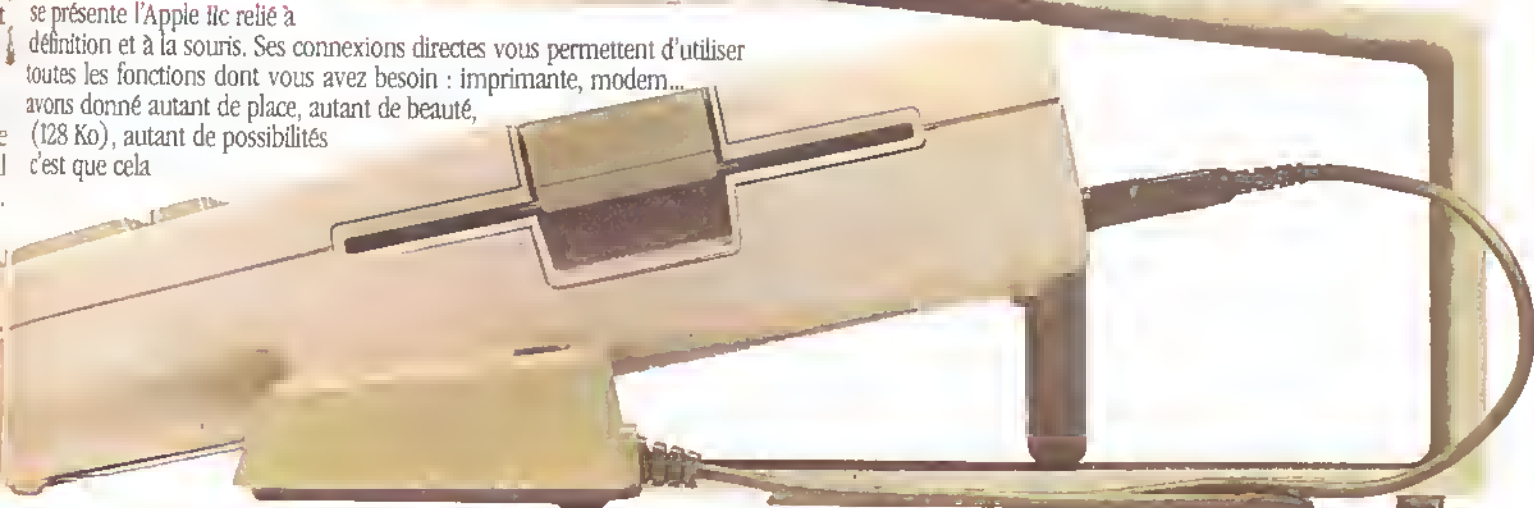
ls au monde ; 16.000
x, traitement de texte,
e, planification.

Voici 8 exemples de ce que vous pouvez afficher sur votre écran. 8 parmi 16.000 ! à vous de découvrir les autres.

Lorsque vous aurez essayé l'Apple IIc vous ne regretterez pas notre générosité.



se présente l'Apple IIc relié à
définition et à la souris. Ses connexions directes vous permettent d'utiliser
toutes les fonctions dont vous avez besoin : imprimante, modem...
avons donné autant de place, autant de beauté,
(128 Ko), autant de possibilités
c'est que cela



Apple présente l'Apple IIc.



Où est la souris ?

Marianne Sutz



Dans un programme Basic, déterminer la présence, ou la non présence, de la souris dans une zone donnée conduit invariablement à des séries de tests aussi peu rapides qu'inélegants. Pour palier cela, nous vous proposons une routine en langage machine d'emploi aisé.

Utilisation de la routine

Il faut tout d'abord implanter la routine en mémoire (dans un tableau de variables entières). Pour cela, vous pouvez employer la méthode qui vous convient le mieux :

- code objet sous forme de "Data" comme dans l'exemple ci-joint;
- code objet sous la forme d'une chaîne de caractères, comme pour les routines "BSAVE et BLOAD" du numéro 17 de Pom's;
- chargement direct en mémoire avec la routine "BLOAD", à condition d'avoir préalablement transformé le code objet en un fichier binaire. Le petit programme suivant se charge de cette tâche :

```
DEFINT A-Z
DIM C(19)
DATA 0,0,0,&h4E56,0,&h487A
DATA &hFFF6,&hA972,&h4267
DATA &h47FA,&hFFEE,&h2F13
DATA &h486E,8,&hA8AD
DATA &h47FA,&hFFE0,&h369F
DATA &h4E5E,&h4E75
```

```
FOR I=0 TO 19:READ C(I):NEXT
OPEN "O",1,"Nom de fichier"
AI=VARPTR(C(0))
FOR I=0 TO 39
PRINT#1,CHR$(PEEK(AI+I));
NEXT
CLOSE
```

Vous pouvez ensuite appeler la routine en spécifiant le quatrième élément du tableau de variables comme adresse de base :

```
Adresse! = VARPTR (Tableau%(3))
pour OPTION BASE 0;
Adresse! = VARPTR (Tableau%(4))
pour OPTION BASE 1.
```

La pile étant utilisée comme tampon, il faut passer les paramètres (coordonnées du rectangle) de manière inhabituelle : d'abord X2, suivi de Y2, X1 et enfin Y1. Ainsi un appel de la routine prend la forme : Adresse! X2%, Y2%, X1%, Y1% ou, avec la version 1.0 du Basic Microsoft, CALL Adresse! (X2%, Y2%, X1%, Y1%). Les arguments doivent absolument être des variables entières ou des expressions résultant en des valeurs inférieures ou égales à 32767 (valeur maximale possible sur 16 bits en complément à deux). Tout autre type d'argument provoquerait un décalage dans la pile, suivi d'effets plus ou moins imprévisibles.

Au retour, le premier élément du tableau de variables entières contient un drapeau indiquant si le curseur se trouve (ou ne se trouve pas) dans le

rectangle donné, et ce quelle que soit la position du bouton de la souris :

```
Tableau%(0) = &h0100 si oui;
Tableau%(0) = &h0000 si non.
```

Exemple d'utilisation

Le petit programme Basic proposé avec cet article illustre une des applications possibles de la routine. On affiche deux rectangles (on pourrait très bien s'en passer, il s'agit simplement de visualiser les zones) puis nous allons voir si le curseur se trouve dans l'un d'eux. On initialise le curseur type "Multiplan" si l'on est dans le rectangle de gauche, le curseur en forme de montre si l'on est dans le rectangle de droite. Dans le cas où nous nous trouvons dans aucun des deux rectangles, on reprend le curseur standard en forme de flèche. Vous pourrez remarquer que le programme arrive à "suivre" des déplacements très rapides de la souris, ce qui ne serait pas le cas avec des tests du type :

```
N%=MOUSE(0)
X%=MOUSE(1)
Y%=MOUSE(2)
IF X%>249 AND X%<451 AND
Y%>49 AND Y%<201 THEN...
```

La routine employée pour changer rapidement la forme du curseur est reprise du numéro précédent de Pom's; nous vous proposons donc de vous y reporter pour de plus amples commentaires.

Source de la routine "Souris/rectangle"

0000	.Trap	_GetMouse	\$A972	; _PtnInRect retourne \$0100 au sommet de la pile si la souris
0000	.Trap	_PtnInRect	\$A8AD	; se trouve à l'intérieur du rectangle donné, ou \$0000 dans le
0000				; cas contraire.
0000	0000	Drapeau	DC	0 ; 2 octets pour passer au Basic le résultat du test.
0002				
0002	0000 0000	Position	DC.L	0 ; 4 octets pour stockage position de la souris (retournée
0006				; par _GetMouse).
0006	4E56 0000	LINK	A6,#0	; A6 contient l'adresse du sommet de la pile.
000A	487A FFF6	PEA	Position	; Empile l'adresse du tampon de 4 octets 'Position'.
000E	A972	_GetMouse		; Retourne les coordonnées de la souris dans 'Position'.
0010	4267	CLR	-(SP)	; 2 octets au sommet de la pile pour le résultat du test.
0012	47FA FFE0	LEA	Position,A3	; L'adresse du tampon 'Position' est placée dans A6.
0016	2F13	MOVE.L	(A3),-(SP)	; Empile le contenu du tampon 'Position'.
0018	486E 0008	PEA	8(A6)	; Empile l'adresse des coordonnées du rectangle (placée dans
001C	A8AD	_PtnInRect		; la pile depuis le Basic).
001E	47FA FFE0	LEA	Drapeau,A3	; L'adresse du tampon 'Drapeau' est placée dans A3.
0022	369F	MOVE	(SP)+,(A3)	; Dépile le résultat du test et le place dans 'Drapeau', qui
0024	4E5E	UNLK	A6	; correspond à l'élément 0 du tableau de variables entières.
0026	4E75	RTS		; Retour au Basic.



Exemple d'utilisation de la routine "souris/rectangle"

DEFINT A-Z

DIM C(19),D(10):Dr=0:DrC=0:Bl=0

' Data pour routine "Souris/rectangle"

DATA 0,0,0,&h4E56,0,&h487A,&hFFF6,&hA972,&h4267

DATA &h47FA,&hFFEE,&h2F13,&h486E,8,&hA8AD

DATA &h47FA,&hFFE0,&h369F,&h4E5E,&h4E75

' Data pour changement de curseur (voir Pom's 18)

DATA &h4E56,0,&h42A7,&h3F2E,8,&hA9B9,&h2057

DATA &h2E90,&hA851,&h4E5E,&h4E75

' Implantation des routines en mémoire

FOR l=0 TO 19:READ C(l):NEXT

FOR i=0 TO 10:READ D(i):NEXT

' Pour visualiser les rectangles

LINE(50,50)-(200,200),,B

LINE(250,50)-(450,200),,B

' Adresses des points d'entrée dans les routines

A!=VARPTR(c(3)):routine "souris/rectangle"

B!=VARPTR(d(0)):routine "curseurs"

Boucle:

Dr=0

' 1 dans Dr si la souris est dans le rectangle de gauche

' 2 dans Dr si la souris est dans le rectangle de droite

AI 200,200,50,50:IF C(0) THEN Dr=1

AI 450,200,250,50:IF C(0) THEN Dr=2

' Si Dr=Drc, on a pas changé de zone

IF Dr=Drc THEN Boucle

Drc=Dr: ON Dr GOTO ID3,ID4

' Si la souris ne se trouve pas dans un rectangle (Dr=0),

' le curseur en forme de flèche est utilisé.

INITCURSOR:GOTO Boucle

' Curseur type "Multiplan" si dans rectangle de gauche

ID3:

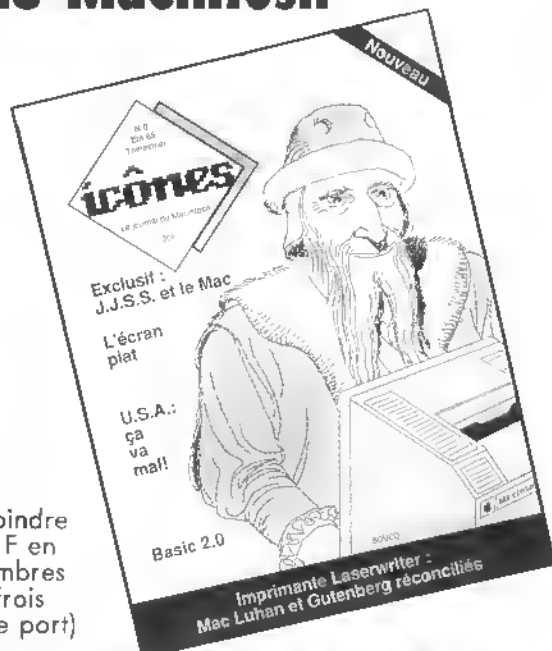
BI 3:GOTO Boucle

' Curseur en forme de montre si rectangle de droite

ID4:

BI 4:GOTO Boucle

Recevez gratuitement le n°0 du journal du Macintosh



Joindre
5 F en
timbres
(Frais
de port)

ICônes : 135 bis, rue du Fg de Roubaix
59800 Lille - Tél. 20.06.30.37

Disquette Macintosh 14/15/16

Cette disquette regroupe les programmes Basic publiés dans les numéros 14, 15 et 16 de Pom's ("Editeur", "Mac/Apple II", "Paint/Basic", "Paint/Start", et d'autres programmes "exemples" employés pour l'illustration des articles publiés dans ces numéros). Vous y trouverez aussi les polices "Los Angeles", "Cairo", "Mos Eisley", "Hollywood" et "Manhattan".

Enfin, les programmes "Localizer" (configuration du clavier AZERTY pour la France ou un autre pays) et "Copie de disque", qui autorise des copies de disquettes avec un seul lecteur sur un Mac 128Ko, en seulement quatre passages.

150,00 F TTC franco
Bon de commande page 74

Pom's

la revue francophone
des utilisateurs du
Macintosh

Lucy : un accessoire de bureau

Jean-Luc Bazanegue



Ici commence une série d'accessoires de bureau destinés à rendre la vie des utilisateurs du Macintosh un peu plus facile. Le premier permet de savoir quelles polices sont disponibles sur la disquette d'amorçage, et ce sans avoir à charger une application. De même, lors de l'emploi d'un logiciel comme MacWrite, Word, etc... il n'est pas particulièrement aisé de savoir dans quelle police se trouve tel caractère spécial; une option de l'accessoire autorise une visualisation de toute la police.

Ceux d'entre-vous qui désirent programmer le Macintosh en assembleur trouveront dans le source de nombreux renseignements sur les fonctions classiques de l'appareil, à savoir :

Traitement de texte

L'accessoire utilise pratiquement tout ce que l'on peut trouver dans un traitement de texte : insertions de caractères ou de chaînes, visualisation du point d'insertion (petite barre clignotante), sélection normale ou étendue (avec "Majuscule-click"), sélection par programme (sans intervention de l'utilisateur), couper, copier, coller, effacer et, enfin, défilement vertical du texte en fonction de la hauteur des caractères. A noter que l'accessoire ne supporte pas - du moins pas encore - les "couper, copier, coller" vers ou depuis une application.

Fenêtres

Le listing montre comment une fenêtre est créée et mise à jour en cas d'effacement partiel ou total suite à un passage au second plan, ou encore à un positionnement hors des limites de l'écran.

Contrôles

Nous metrons ici en oeuvre un bouton standard, un bouton type "radio", des cases de contrôle et enfin des barres de défilement. Ces dernières sont un peu plus complexes à utiliser que les contrôles précédents, et nous pensons que le source éclaircit certains points portant sur la gestion de ces fonctions essentielles.

Fonctions diverses

Nous employons aussi diverses routines liées aux polices de caractères, à la souris, à l'affichage de dessins, à la conversion de nombres en chaînes, etc... Cet accessoire sans prétention représente ainsi un concentré de ce que l'on peut trouver dans une application plus ambitieuse.

Faute de place, nous n'avons pas pu joindre au listing toutes les remarques nécessaires à la bonne compréhension du système. En revanche, pour ceux qui veulent en savoir plus, la disquette Mac 19 comporte, outre l'accessoire en question (que vous pourrez installer sur d'autres disquettes avec "Font/DA Mover"), le même source mais cette fois commenté point par point. Celui-ci peut bien sûr être assemblé ou consulté avec l'éditeur du système de développement 68000, mais peut aussi être chargé depuis "DiskWrite".

Voici pour ce premier accessoire, mais nous avons déjà plusieurs idées en tête. Par exemple : ne vous est-il jamais arrivé de vous dire, lorsque le Mac vous transmet un message du type "La disquette est presque saturée...", qu'il serait pratique de pouvoir détruire ou protéger un fichier sans quitter l'application en cours ? Enfin, vous désirez peut être savoir pourquoi nous avons baptisé la chose "Lucy in the Sky with Diamonds". Eh bien... pourquoi pas ?

Relations entre accessoires de bureau et applications

Un accessoire de bureau peut être considéré comme une "mini-application" capable de fonctionner en parallèle avec l'application en cours. Pour que ceci puisse se faire, l'application doit suivre des règles précises.

Tout d'abord, l'application doit afficher le menu "Pomme" afin de permettre à l'utilisateur l'appel d'un accessoire. Elle doit aussi comporter le menu "Edition", avec les commandes "Couper, Copier, Coller, Effacer", en prévision d'éventuels accessoires utilisant ces fonctions. Un accessoire de bureau peut avoir son

propre menu ; une application doit donc réserver, dans la barre des menus, la place pour un menu supplémentaire (si l'on regarde un programme comme MacDraw, on s'aperçoit que la règle n'est pas toujours respectée !).

Pendant le déroulement de l'application, et si l'utilisateur choisit un accessoire dans le menu "Pomme", l'application doit se charger de l'ouverture de l'accessoire. Lorsque l'accessoire est ouvert, à chaque fois que l'utilisateur appuie sur le bouton de la souris, l'application doit déterminer si l'action est effectuée dans l'accessoire, auquel cas elle doit lui transmettre l'événement par l'intermédiaire d'une routine du système. Cet événement sera ensuite traité par l'accessoire lui-même. Si plusieurs accessoires sont présents au même moment sur le bureau, le principe est le même, l'événement étant simplement dirigé vers l'accessoire concerné. Lorsque l'utilisateur passe d'une fenêtre de l'application à la fenêtre d'un accessoire, l'application doit rendre invalides les articles des menus qui lui sont propres et rendre valides les articles "Couper, Copier, Coller et Effacer" du menu "Edition". L'inverse doit se produire au moment où l'utilisateur revient à l'application (l'état du menu "Edition" au retour d'un accessoire de bureau est déterminé par l'application).

La fermeture de l'accessoire peut s'effectuer de différentes façons :

- si l'article "Fermer" est disponible dans un menu "Fichier", et que l'utilisateur choisit cet article au moment où l'accessoire est actif, l'application doit se charger de la fermeture de l'accessoire ; si l'accessoire comporte une case de fermeture, et que l'utilisateur "clique" sur cette case, le gestionnaire du bureau (une partie du système d'exploitation du Mac) se charge de la fermeture ;
- si l'utilisateur quitte l'application en cours, le ou les accessoires sont automatiquement fermés par le système ;
- enfin, un accessoire peut très bien se fermer lui-même.

Lucy.Asm

; Routines utilisées par
; l'accessoire. Sur un 512K,
; la définition des 'Traps'
; peut être remplacée par ;
; INCLUDE MacTraps.D
; *****

; Gestionnaire de ressources

```
.Trap _CountResources $A99C
.Trap _GetIndResource $A99D
.Trap _GetResInfo $A9A9
```

; QuickDraw

```
.Trap _CopyBits $A8EC
.Trap _DrawString $A864
.Trap _EraseRect $A8A3
.Trap _FrameRect $A8A1
.Trap _GetFontInfo $A88B
.Trap _GlobalToLocal $A871
.Trap _MoveTo $A893
.Trap _PInRect $A8A0
.Trap _SetPort $A873
.Trap _TextFace $A889
.Trap _TextFont $A887
.Trap _TextSize $A88A
```

; Gestionnaire de polices

```
.Trap _RealFont $A902
```

; Gestionnaire d'événements

```
.Trap _GetMouse $A972
```

; Gestionnaire de fenêtres

```
.Trap _BeginUpdate $A922
.Trap _DisposeWindow $A914
.Trap _EndUpdate $A923
.Trap _NewWindow $A913
```

; TEXTEdit

```
.Trap _TEActivate $A9D8
.Trap _TEDeactivate $A9D9
.Trap _TECAtText $A9D0
.Trap _TEClick $A9D4
.Trap _TECopy $A9D5
.Trap _TECut $A9D6
.Trap _TEDelete $A9D7
.Trap _TEDispose $A9CD
.Trap _TEIdle $A9DA
.Trap _TEInsert $A9DE
.Trap _TEKey $A9DC
.Trap _TENew $A9D2
.Trap _TEPaste $A9DB
.Trap _TEScroll $A9DD
.Trap _TESetSelect $A9D1
```

```
.Trap _TEUpdate $A9D3
```

; Gestionnaire de contrôles

```
.Trap _DrawControls $A969
.Trap _FindControl $A96C
.Trap _GetMaxCtl $A962
.Trap _GetCtlValue $A960
.Trap _GetCRefCon $A95A
.Trap _NewControl $A954
.Trap _SetCtlValue $A963
.Trap _SetMaxCtl $A965
.Trap _SetMinCtl $A964
.Trap _TrackControl $A968
```

; Utilitaires 'ToolBox'

```
.Trap _Pack7 $A9EE
```

; Equivalences. Sur un 512K,

; elles peuvent être

; remplacées par :

; INCLUDE SysEqu.D

; INCLUDE ToolEqu.D

; INCLUDE QuickEqu.D

```
mButDwnEvt EQU 1
keyDwnEvt EQU 3
keyUpEvt EQU 4
autoKeyEvt EQU 5
updatEvt EQU 9
noGrowDocProc EQU 4
teLineHile EQU $19
teAscent EQU $1A
teFont EQU $4A
teFace EQU $4C
taMode EQU $4E
teSize EQU $50
teNLines EQU $5E
teLines EQU $60
dCtlWindow EQU $1E
dCtlRefNum EQU $19
WindowKind EQU $6C
CSCode EQU $1A
CSPParam EQU $1C
evtNum EQU $0
evtMessage EQU $2
evtMouse EQU $A
evtMete EQU $E
evtMBut EQU $F
checkBoxProc EQU 1
radioButProc EQU 2
scrollBarProc EQU 19
accEvent EQU $40
accRun EQU $41
accUndo EQU $44
accCut EQU $46
```

```
accCopy EQU $47
accPaste EQU $48
accClear EQU $49
inUpButton EQU 20
inDownButton EQU 21
inPageUp EQU 22
inPageDown EQU 23
inThumb EQU 129
```



; Début du code. Un accessoire de bureau est un
; 'Driver'. A ce titre, il comporte un 'en-tête'
; qui n'apparaît pas dans une application
; *****

```
Base DC $2400
DC 10
DC $016A
DC 0
DC Ouverture-Base
DC Ouverture-Base
DC Controle-Base
DC Ouverture-Base
DC Fermeture-Base
```

```
Titre DC.B 29,'Lucy in the Sky with Diamonds'
```

; Ouverture de l'accessoire : affichage de la
; fenêtre, affichage des boutons, etc...
; *****

; Fenêtre existante ? Si non, mise en place de
; l'accessoire

```
Ouverture MOVE.M A1-A4,-(SP)
MOVE.L A1,A4
TST.L D0,Window(A4)
BNE FenetreEx
```

; Initialisation de la fenêtre

```
CLR.L -(SP)
CLR.L -(SP)
PEA RectDial
PEA Titre
MOVE #$100,-(SP)
MOVE #noGrowDocProc,-(SP)
MOVE.L #1,D0
MOVE.L D0,-(SP)
MOVE #$100,-(SP)
CLR.L -(SP)
_NewWindow
MOVE.A.L (SP),A0
MOVE.L A0,DctlWindow(A4)
MOVE DctlRefNum(A4),WindowKind(A0)
MOVE.L A0,-(SP)
_SetPort
```

; Initialisation du tampon pour police complète

Disquette Macintosh numéro 17

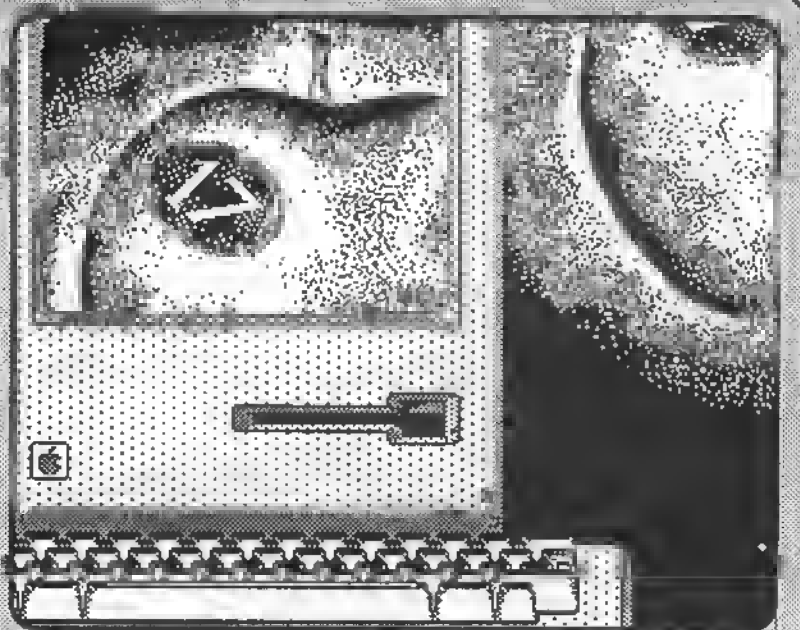
La Disquette Mac 17 comporte les
programmes Basic :

"Catalogue sur imprimante", pour obtenir
toutes les informations sur les fichiers
(type, créateur, index, position du premier
bloc pour la partie "données", position du
premier bloc pour la partie "ressource",
longueurs logique et physique, fichiers
protégés ou non, icônes invisibles, etc...)
sur papier ou à l'écran.

Routines "BLOAD" et "BSAVE", similaires
aux routines de l'Apple II, elles autorisent
des sauvegardes et chargements rapides de
fichiers binaires (données graphiques,
tableaux de variables, etc...).

80,00 F TTC franco

Bon de commande page 74





```
LEA    TamponPolice,A6
MOVE   #21,D7
PoliceComp MOVE.B D7,(A6)+
MOVE.B #20,(A6)+
ADDO   #1,D7
CMPI   #256,D7
BNE.S  PoliceComp

; Début mise à jour
MOVE.L DCtWindow(A4),-(SP)
_BaginUpdate

; Vers sous-programme affichage des rectangles
BSR    Rectangles

; Vers sous-programme affichage style polices
CLR    -(SP)
_TextFont
BSR    Chaînes

; Vers sous-programme affichage 'bestiole'
MOVE.L DCtWindow(A4),D7
BSR    AtCoc

; Réinitialisation style
LEA    Apparence,A6
CLR    (A6)

; Initialisation des zones de contrôle
; Barre de défilement pour le texte
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectCil1
CLR.L  -(SP)
MOVE   #100,-(SP)
CLR.L  -(SP)
MOVE   #48,-(SP)
MOVE   #scrollBarProc,-(SP)
CLR.L  -(SP)
_NewControl
LEA    HandleCil1,A6
MOVE.L (SP)+,(A6)

; Barre de défilement pour choix police
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectCil2
CLR.L  -(SP)
MOVE   #100,-(SP)
CLR    -(SP)
CLR.L  -(SP)
MOVE   #scrollBarProc,-(SP)
MOVEQ.L #1,D7
MOVE.L D7,-(SP)
_NewControl
LEA    HandleCil2,A6
MOVE.L (SP)+,(A6)

; Barre de défilement pour choix taille
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectCil3
CLR.L  -(SP)
MOVE   #100,-(SP)
CLR    -(SP)
CLR.L  -(SP)
MOVE   #scrollBarProc,-(SP)
MOVEQ.L #2,D7
MOVE.L D7,-(SP)
_NewControl
LEA    HandleCil3,A6
MOVE.L (SP)+,(A6)

; Bouton 'OK'
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectOK
PEA    NomOK
MOVE   #100,-(SP)
CLR.L  -(SP)
CLR.L  -(SP)
MOVEQ.L #3,D7
MOVE.L D7,-(SP)
_NewControl
LEA    HandleOK,A6
MOVE.L (SP)+,(A6)

; Bouton 'Standard'
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectStand
CLR.L  -(SP)
MOVE   #100,-(SP)
MOVE   #1,-(SP)
CLR    -(SP)
MOVE   #1,-(SP)
```

```
MOVE   #radioButProc,-(SP)
MOVEQ.L #4,D7
MOVE.L D7,-(SP)
_NewControl
LEA    HandleStand,A6
MOVE.L (SP)+,(A6)

; Bouton 'Gras'
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectGras
CLR.L  -(SP)
MOVE   #100,-(SP)
CLR.L  -(SP)
MOVE   #1,-(SP)
MOVEQ.L #5,D7
MOVE   #checkBoxProc,-(SP)
MOVE.L D7,-(SP)
_NewControl
LEA    HandleGras,A6
MOVE.L (SP)+,(A6)

; Bouton 'Italique'
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectItal
CLR.L  -(SP)
MOVE   #100,-(SP)
CLR.L  -(SP)
MOVE   #1,-(SP)
MOVE   #checkBoxProc,-(SP)
MOVEQ.L #6,D7
MOVE.L D7,-(SP)
_NewControl
LEA    HandleItal,A6
MOVE.L (SP)+,(A6)

; Bouton 'Souligné'
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectSoul
CLR.L  -(SP)
MOVE   #100,-(SP)
CLR.L  -(SP)
MOVE   #1,-(SP)
MOVE   #checkBoxProc,-(SP)
MOVEQ.L #7,D7
MOVE.L D7,-(SP)
_NewControl
LEA    HandleSoul,A6
MOVE.L (SP)+,(A6)

; Bouton 'Relief'
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectRelief
CLR.L  -(SP)
MOVE   #100,-(SP)
CLR.L  -(SP)
MOVE   #1,-(SP)
MOVE   #checkBoxProc,-(SP)
MOVEQ.L #6,D7
MOVE.L D7,-(SP)
_NewControl
LEA    HandleRelief,A6
MOVE.L (SP)+,(A6)

; Bouton 'Ombre'
CLR.L  -(SP)
MOVE.L DCtWindow(A4),-(SP)
PEA    RectOmbre
CLR.L  -(SP)
MOVE   #100,-(SP)
CLR.L  -(SP)
MOVE   #1,-(SP)
MOVE   #checkBoxProc,-(SP)
MOVEQ.L #9,D7
MOVE.L D7,-(SP)
_NewControl
LEA    HandleOmbre,A6
MOVE.L (SP)+,(A6)

; .....
; Affichage des polices présentes sur la
; disquette (dans le fichier système).
; .....
; Initialisation de la zone de texte
CLR    -(SP)
_TextFace
CLR.L  -(SP)
PEA    RectVPolices
PEA    RectVPolices
_NEW
```

```
LEA    HandlePolices,A6
MOVE.L (SP)+,(A6)
; Nombre de ressources de type 'FONT'
CLR    -(SP)
MOVE.L #464F4E54,-(SP)
_CountResources
MOVE   (SP)+,D7

; Lecture et affichage des noms
LEA    NumRetPolices,A3
LEA    BGetDItem,A2
CLR    D6
CLR.L  -(SP)
BLNom MOVE.L #464F4E54,-(SP)
MOVE   D7,-(SP)
_GetIndResource
PEA    BGetDItem
PEA    BgetDItem+2
PEA    TamponNom
_GetResourceInfo
LEA    TamponNom,A6
MOVE.B (A6),D3
BEQ    BErr
MOVE.B #32,(A6)
ADDO   #1,D6
MOVE   (A2),(A3)+
MOVE.L A6,-(SP)
AND.L  #FFF,D3
ADDO   #1,D3
MOVE.L D3,-(SP)
LEA    HandlePolices,A6
MOVE.L (A6),-(SP)
_Insert
PEA    CR
MOVE.L #2,-(SP)
MOVE.L (A6),-(SP)
_Insert
PEA    SUBQ
SUBQ   #1,D7
BNE.S  BLNom

; Valeur maxi de la barre de défilement
; en fonction du nombre de polices
LEA    NombPolices,A8
MOVE   D6,(A6)+
MOVE   #1,(A6)
LEA    HandlePolices,A3
MOVE.L (A3),-(SP)
_Activate
CMP    #6,-2(A8)
BLE.S  Retour
LEA    HandleCil2,A3
MOVE.L (A3),-(SP)
MOVE   -2(A8),D7
SUBQ   #6,D7
MOVE   D7,-(SP)
_SetMaxCil
Retour LEA    HandlePolices,A6
MOVEQ   #1,D7

; Vers sous-programme pour visualiser
; la police sélectionnée
BSR    Selection

; Vers sous-programme pour affichage
; tailles valides
BSR    TaillesP

; Vers sous-programme caractéristiques
; de la police
BSR    CaracTexte

; Initialisation zone pour le texte et
; affichage du texte par défaut
CLR.L  -(SP)
PEA    RectDTexte
PEA    RectVTexte
_NEW
LEA    HandleTexte,A6
MOVE.L (SP)+,(A6)
PEA    NomDefaut
MOVE.L #87,-(SP)
MOVE.L (A6),-(SP)
_Insert

; Fin mise à jour
MOVE.L DCtWindow(A4),-(SP)
_EndUpdate

; Activation de la zone de texte
LEA    HandleTexte,A6
MOVE.L (A6),-(SP)
_Activate

; Fin de la mise en place de l'accessoire
FenetreEx MOVE.L (SP)+,A1-A4
```



```

CLR      D0
RTS

; Routine appelée par le système en cas
; d'événement correspondant au masque situé
; dans l'entête de l'accessoire
;
Controle  MOVEM.L A1-A4,-(SP)
          MOVE.L  A1,A4
          MOVE    CScode(A0),D0
          CMPI    #accEvent,D0
          BEQ     Evenements
          CMPI    #accRun,D0
          BEQ     Tempo
          CMPI    #accCut,D0
          BEQ     Couper
          CMPI    #accCopy,D0
          BEQ     Copier
          CMPI    #accPaste,D0
          BEQ     Colier
          CMPI    #accClear,D0
          BEQ     Effacer
FinControle  MOVEM.L (SP)+,A1-A4
          CLR     D0
          RTS

; Fermeture de l'accessoire
;
Fermeture  MOVEM.L A1-A4,-(SP)
          MOVE.L  A1,A4
          MOVE.L  DCtiWindow(A4),-(SP)
          CLR.L   DCtiWindow(A4)
          DispsWindow
          BRA.S   FinControle

; Routine appelée six fois par seconde pour
; le clignotement de la barre d'insertion
;
Tempo      LEA     HandleTexte,A6
          MOVE.L  (A6),-(SP)
          TEIdle
          BRA     FinControle

; Traitement des événements
;
Evenements  MOVEM.L CSPParam(A0),A4
          MOVE    EvtNum(A4),D0
          CMPI    #mButDwnEvt,D0
          BNE.S   Ev1
          BSR     Contenu
          BRA     FinControle

Ev1         CMPI    #keyDwnEvt,D0
          BEQ     ToucheE
          CMPI    #autoKeyEvt,D0
          BEQ     ToucheE
          CMPI    #updatEvt,D0
          BNE     Ev2
          BSR     MiseJour
          BRA     FinControle

Ev2         ToucheE  MOVE.B  evtMotif(A4),D1
          ANDI.B   #1,D1
          BEQ.S    ToucheE0
          MOVE.B   evtMessage+2(A4),D0
          CMPI.B   #8,D0
          BEQ     Couper
          CMPI.B   #9,D0
          BEQ     Copier
          CMPI.B   #11,D0
          BEQ     Colier
          MOVE.L   evtMessage(A4),D0
          ANDI     #$FF,D0
          MOVE     D0,-(SP)
          LEA      HandleTexte,A4
          MOVE.L   (A4),-(SP)
          TEKey
          BRA     FinControle

ToucheE0    ToucheE  MOVE.L  EvtMessage(A4),-(SP)
          _BeginUpdate
          MOVE.L   EvtMessage(A4),-(SP)
          _SetPort
          BSR     Rectangles
          CLR     -(SP)
          _TextFont
          CLR     -(SP)
          _TextFace
          MOVE    #12,-(SP)
          _TextSize
          BSR     Chaines
          MOVE.L  EvtMessage(A4),D7
          BSR     AICoc
          PEA     RectVTexte
          LEA     HandleTexte,A6
          MOVE.L  (A6),-(SP)
          TEUpdate
          PEA     RectVPolices
          LEA     HandlePolices,A6

Contenu     LEA     DialogPtr,A6
          MOVE.L   DCtiWindow(A1),(A6)
          MOVE.L   (A6),-(SP)
          _SetPort
          LEA     EventRecord,A6
          MOVEQ    #3,D7
          CLR     D6
          MOVE.L   0(A4,D6),0(A6,D6)
          ADDQ     #4,D6
          DBRA     D7,TrEvR
          PEA     EventRecord+10
          _GlobeToLocal
          CLR.B    -(SP)
          LEA     EventRecord+10,A6
          MOVE.L   (A6),-(SP)
          PEA     RectVPolices
          _PtlInRect
          MOVE.B   (SP)+,D0
          BNE     DFPolices
          CLR.B    -(SP)
          LEA     EventRecord+10,A6
          MOVE.L   (A6),-(SP)
          PEA     RectVTailles
          _PtlInRect
          MOVE.B   (SP)+,D0
          BNE     DFTailles
          CLR.B    -(SP)
          LEA     EventRecord+10,A6
          MOVE.L   (A6),-(SP)
          PEA     RectVTexte
          _PtlInRect
          MOVE.B   (SP)+,D0
          BNE     DFTexte
          CLR.B    -(SP)
          LEA     EventRecord+10,A6
          MOVE.L   (A6),-(SP)
          PEA     RectCoc
          _PtlInRect
          MOVE.B   (SP)+,D0
          BNE     DCoc

TrEvR       LEA     DialogPtr,A6
          MOVE.L   DCtiWindow(A1),(A6)
          MOVE.L   (A6),-(SP)
          _SetPort
          LEA     EventRecord,A6
          MOVEQ    #3,D7
          CLR     D6
          MOVE.L   0(A4,D6),0(A6,D6)
          ADDQ     #4,D6
          DBRA     D7,TrEvR
          PEA     EventRecord+10
          _GlobeToLocal
          CLR.B    -(SP)
          LEA     EventRecord+10,A6
          MOVE.L   (A6),-(SP)
          PEA     RectVPolices
          _PtlInRect
          MOVE.B   (SP)+,D0
          BNE     DFPolices
          CLR.B    -(SP)
          LEA     EventRecord+10,A6
          MOVE.L   (A6),-(SP)
          PEA     RectVTailles
          _PtlInRect
          MOVE.B   (SP)+,D0
          BNE     DFTailles
          CLR.B    -(SP)
          LEA     EventRecord+10,A6
          MOVE.L   (A6),-(SP)
          PEA     RectVTexte
          _PtlInRect
          MOVE.B   (SP)+,D0
          BNE     DFTexte
          CLR.B    -(SP)
          LEA     EventRecord+10,A6
          MOVE.L   (A6),-(SP)
          PEA     RectCoc
          _PtlInRect
          MOVE.B   (SP)+,D0
          BNE     DCoc

; 'Click' dans la fenêtre de l'accessoire
; Routine pour déterminer la zone concernée
;

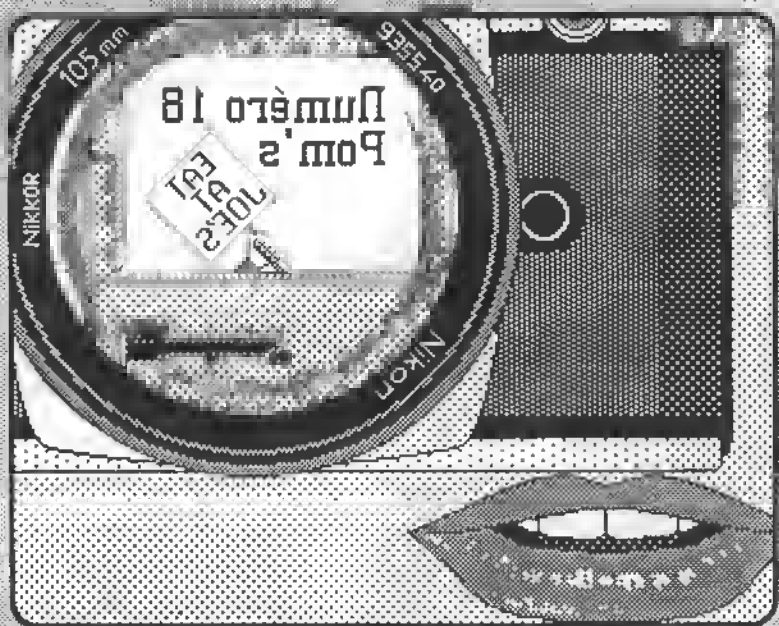
```



Disquette Macintosh numéro 18

Elle comporte le nouveau fichier système et la version 4.1 du Finder, plus rapide et complète que la précédente. Le programme "System update" autorise l'implantation des nouveaux Finder et Système sur vos anciennes disquettes. "BR Démo" est un programme qui permet l'affichage temporisé d'écrans créés avec la séquence de touches "Commande-Shift-Majuscule-3". "Font/DA Mover" est une version améliorée de "Font Mover". De plus, il permet de manipuler les accessoires de bureau. Enfin, outre les programmes publiés dans le numéro 18, vous trouverez les polices Times, Courier, Symbol et Helvetica.

80,00 F TTC franco
Bon de commande page 74



```

; Routine pour déterminer si action dans
; une zone de contrôle et, si oui, laquelle

```

```

CLR      -(SP)
MOVE.L   (A6),-(SP)
LEA      DialogPtr,A4
MOVE.L   (A4),-(SP)
PEA      WhichControl
_FindControl
MOVE     (SP)+,D0
BEQ.S    ContS
LEA      WhichControl,A3
LEA      ValeurCtl,A2
CLR      -(SP)
MOVE.L   (A3),-(SP)
_GetCtlValue
MOVE     (SP)+,(A2)
CLR      -(SP)
MOVE.L   (A3),-(SP)
MOVE.L   (A6),-(SP)
CLR.L    -(SP)
_TrackControl
MOVE     (SP)+,D0
BNE.S    Cont1
RTS
ContS
Cont1
CLR.L    -(SP)
MOVE.L   (A3),-(SP)
_GetCRefCon
MOVE.L   (SP)+,D1
BEQ      ControlTexta
CMP#     #1,D1
BEQ      ControlP
CMP#     #2,D1
BEQ      ControlTail
CMP#     #3,D1
BEQ      BOK
CMP#     #4,D1
BEQ      BStd
BRA      BApparence

```

```

; Traitement d'une action dans la barre de
; défilement des polices ou des tailles

```

```

ControlP  CLR      D4
          BRA.S    ControlPT0
ControlTail MOVEQ    #4,D4
ControlPT0 CMP#     #254,D0
          BEQ      ContS
          CMP#     #inUpButton,D0
          BEQ      FlecheHP
          CMP#     #inDownButton,D0
          BEQ      FlecheBP
          CMP#     #inPageUp,D0
          BEQ      PageHautP
          CMP#     #inPageDown,D0
          BEQ      PageBasP
          CMP#     #inThumb,D0
          BEQ      AscenseurP
          BRA      ContS

```

```

; Traitement d'une action sur l'ascenseur
; de la barre de défilement des polices ou
; des tailles

```

```

AscenseurP CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetCtlValue
          MOVE     (SP)+,D5
          LEA      ValeurCtl,A4
          MOVE     (A4),D6
          SUB      D5,D6
          MOVE     #16,D7
          MULU     D6,D7
          CLR      -(SP)
          MOVE     D7,(SP)
          LEA      HandlePolices,A3
          MOVE.L   0(A3,D4),-(SP)
          _TEScroll
          RTS

```

```

; Traitement d'une action sur la partie
; "page haut" de la barre de défilement des
; polices ou des tailles

```

```

PageHautP CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetCtlValue
          MOVE     (SP)+,D5
          MOVE     D5,D6

```

```

SUBQ     #6,D6
BMI.S    PageHautP0
SUBQ     #8,D5
MOVE     #16,D6
BRA.S    PageHautP1
PageHautP0 MOVE     #16,D6
          MULU     D5,D6
          CLR      D5
          CLR      -(SP)
          MOVE     D6,(SP)
          LEA      HandlePolices,A6
          MOVE.L   0(A6,D4),-(SP)
          _TEScroll
          MOVE.L   (A3),-(SP)
          MOVE     D5,(SP)
          _SetCtlValue
          RTS

```

```

; Traitement d'une action sur la partie
; "page bas" de la barre de défilement des
; polices ou des tailles

```

```

PageBasP  CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetCtlValue
          MOVE     (SP)+,D5
          CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetMaxCtl
          MOVE     (SP)+,D5
          MOVE     D5,D7
          ADDQ     #6,D7
          CMP      D6,D7
          BGT.S    PageBasP0
          ADDQ     #6,D5
          MOVE     #96,D6
          BRA.S    PageBasP1
PageBasP0 MOVE     D6,D7
          SUB      D5,D7
          MOVE     D6,D5
          MOVE     #16,D6
          MULU     D7,D5
          CLR      -(SP)
          MOVE     D5,(SP)
          LEA      HandlePolices,A6
          MOVE.L   0(A6,D4),-(SP)
          _TEScroll
          MOVE     D5,(SP)
          _SetCtlValue
          RTS

```

```

; Traitement d'une action sur la flèche
; vers le haut de la barre de défilement
; des polices ou des tailles

```

```

FlecheHP  CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetCtlValue
          MOVE     (SP)+,D0
          BEQ      ContS
          SUBQ     #1,D0
          MOVE.L   (A3),-(SP)
          MOVE     D0,(SP)
          _SetCtlValue
          CLR      -(SP)
          MOVE     #16,(SP)
          LEA      HandlePolices,A3
          MOVE.L   0(A3,D4),-(SP)
          _TEScroll
          RTS

```

```

; Traitement d'une action sur la flèche
; vers le bas de la barre de défilement
; des polices ou des tailles

```

```

FlecheBP  CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetCtlValue
          MOVE     (SP)+,D5
          CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetMaxCtl
          MOVE     (SP)+,D5
          CMP      D6,D5
          BEQ.S    FIB0
          ADDQ     #1,D5
          MOVE.L   (A3),-(SP)

```

```

MOVE     D5,-(SP)
_SetCtlValue
CLR      -(SP)
MOVE     #16,(SP)
LEA      HandlePolices,A3
MOVE.L   0(A3,D4),-(SP)
_FIB0
RTS

```

```

; Action dans la barre de défilement du
; texte

```

```

ControlTexta LEA      HandleTexte,A6
          MOVE.L   (A6),A6
          MOVE.L   (A6),A6
          MOVE     $18(A6),D7
          CMP#     #254,D0
          BEQ      ContS
          CMP#     #inUpButton,D0
          BEQ.S    FlecheH
          CMP#     #inDownButton,D0
          BEQ.S    FlecheB
          CMP#     #inPageUp,D0
          BEQ      PageHaut
          CMP#     #inPageDown,D0
          BEQ      PageBas
          CMP#     #inThumb,D0
          BEQ      Ascenseur
          RTS

```

```

; Traitement d'une action sur la flèche
; vers le haut de la barre de défilement
; du texte

```

```

FlecheH   CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetCtlValue
          MOVE     (SP)+,D0
          BEQ.S    FIO
          SUBQ     #1,D0
          MOVE.L   (A3),-(SP)
          MOVE     D0,(SP)
          _SetCtlValue
          CLR      -(SP)
          MOVE     D7,(SP)
          LEA      HandleTexte,A3
          MOVE.L   (A3),-(SP)
          _TEScroll
          RTS

```

```

; Traitement d'une action sur la flèche
; vers le bas de la barre de défilement
; du texte

```

```

FlecheB   CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetCtlValue
          MOVE     (SP)+,D0
          CMP#     #48,D0
          BEQ.S    FIB0
          ADDQ     #1,D0
          MOVE.L   (A3),-(SP)
          MOVE     D0,(SP)
          _SetCtlValue
          CLR      -(SP)
          NEG      D7
          MOVE     D7,(SP)
          LEA      HandleTexte,A3
          MOVE.L   (A3),-(SP)
          _TEScroll
          RTS

```

```

; Traitement d'une action dans la partie
; "page haut" de la barre de défilement
; du texte

```

```

PageHaut  CLR      -(SP)
          MOVE.L   (A3),-(SP)
          _GetCtlValue
          MOVE     (SP)+,D5
          MOVE.L   #112,D0
          DIVU     D7,D0
          MOVE     D5,D6
          SUB      D0,D6
          BMI.S    PageHaut0
          SUB      D0,D5
          MULU     D0,D7

```



```

MOVE D7,D6
BRA.S PageHaut1
MULU D5,D7
MOVE D7,D6
CLR D6
PageHaut1 CLR -(SP)
MOVE D6,(SP)
LEA HandleTexte,A6
MOVE.L (A6),-(SP)
_TEScroll
MOVE.L (A3),-(SP)
MOVE D5,(SP)
_SetCtlValue
RTS

```

; Traitement d'une action sur l'ascenseur
; de la barre de défilement du texte.

```

Ascenseur CLR -(SP)
MOVE.L (A3),-(SP)
_GetCtlValue
MOVE (SP)+,D5
LEA ValeurCtl,A4
MOVE (A4),D6
SUB D5,D6
MULU D6,D7
CLR -(SP)
MOVE D7,(SP)
LEA HandleTexte,A3
MOVE.L (A3),-(SP)
_TEScroll
RTS

```

; Traitement d'une action dans la partie
; "page bas" de la barre de défilement
; du texte.

```

PageBas CLR -(SP)
MOVE.L (A3),-(SP)
_GetCtlValue

MOVE (SP)+,D5
CLR -(SP)
MOVE.L (A3),-(SP)
_GetMaxCtl
MOVE (SP)+,D6
MOVE.L #112,D0
DIVU D7,D0
MOVE D5,D4
ADD D0,D4
CMP D6,D4
BGT.S PageBas0
ADD D0,D5
MULU D0,D7

```

```

NEG D7
MOVE D7,D6
BRA.S PageBas1
MOVE D6,D4
SUB D0,D4
MOVE D6,D5
MULU D4,D7
NEG D7
MOVE D7,D6
CLR -(SP)
MOVE D6,(SP)
LEA HandleTexte,A6
MOVE.L (A6),-(SP)
_TEScroll
MOVE.L (A3),-(SP)
MOVE D5,(SP)
_SetCtlValue
RTS

```

; Sous-programme pour tracé des rectangles

```

Rectangles PEA RectTexte
_FrameRect
PEA RectPolices
_FrameRect
PEA RectTailles
_FrameRect
RTS

```

; Sous-programme pour affichage des chaînes

```

Chaines MOVE #226,D3
MOVE #142,D4
MOVE #16,D5
MOVE D3,(SP)
MOVE D4,(SP)
_MoveTo
PEA C1
_DrawString
MOVE #1,(SP)
_TextFace
ADD D5,D4
MOVE D3,(SP)
MOVE D4,(SP)
_MoveTo
PEA C2
_DrawString
MOVE #2,(SP)
_TextFace
ADD D5,D4
MOVE D3,(SP)
MOVE D4,(SP)
_MoveTo
PEA C3
_DrawString

```

```

MOVE #4,-(SP)
_TextFace
ADD D5,D4
MOVE D3,-(SP)
MOVE D4,-(SP)
_MoveTo
PEA C4
_DrawString
MOVE #8,-(SP)
_TextFace
ADD D5,D4
MOVE D3,-(SP)
MOVE D4,-(SP)
_MoveTo
PEA C6
_DrawString
MOVE #16,(SP)
_TextFace
ADD D5,D4
MOVE D3,(SP)
MOVE D4,(SP)
_MoveTo
PEA C6
_DrawString
RTS

```



; Sous-programme pour affichage "bestiole"

```

AICoc LEA BitMapCoc,A6
LEA DefaCoc,A3
MOVE.L A3,(A6)
MOVE.L A6,-(SP)
MOVEA.L D7,A6
ADDQ.L #2,A6
MOVE.L A6,-(SP)
PEA RectSrcCoc
PEA RectCoc
CLR -(SP)
CLR.L -(SP)
_CopyBite
RTS

```

; Sous-programme pour visualisation de la
; police ou de la taille sélectionnée

```

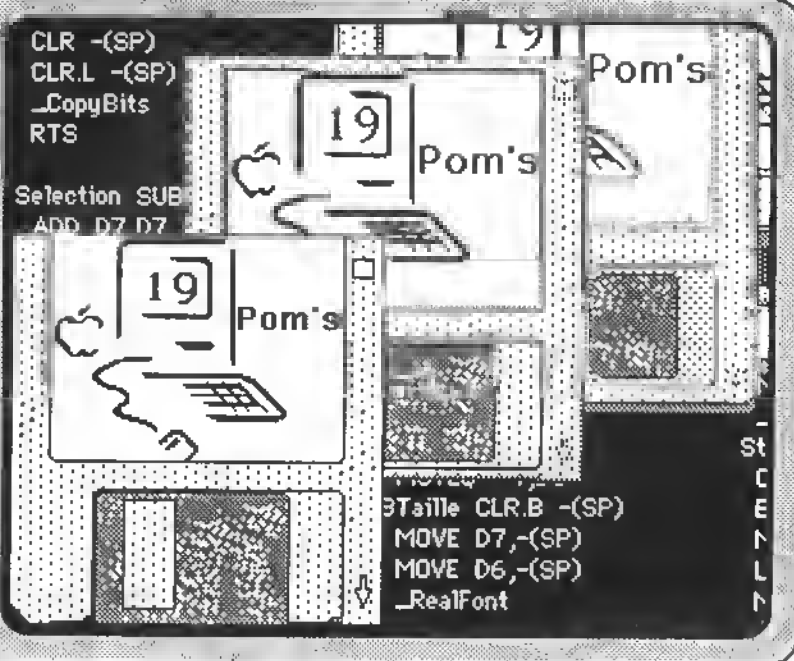
Selection SUBQ #1,D7
ADD D7,D7
MOVEA.L A6,A3
MOVEA.L (A6),A6
CLR.L D6
MOVE teLines(A6,D7),D6
MOVE.L D6,-(SP)
CLR.L D6

```

Disquette Macintosh numéro 19

Vous trouverez, sur cette nouvelle disquette, la nouvelle version de MacWrite (4.5). Ce programme de traitement de texte, contrairement à son prédécesseur, est "Disk Based", ce qui signifie que la taille des fichiers traités n'est plus directement liée à la mémoire disponible. MacWrite 4.5 est accompagné de MacPaint 1.5, qui a l'avantage de ne plus "perdre la grille" lors des déplacements. Le dossier système comporte les accessoires de bureau "Lucy" et "5" (le fichier source de "Lucy", situé sur la disquette, est commenté point par point). Les exemples liés aux articles sont bien sûr présents sur la disquette.

80,00 F TTC franco
Bon de commande page 74



```

MOVE    teLines+2(A6,D7),D6
MOVE.L  D6,-(SP)
MOVE.L  (A3),-(SP)
_TESetSelect
RTS

```

```

; Sous-programme pour détermination et
; affichage des tailles valides

```

```

TaillesP  MOVEM.L  A4,-(SP)
          CLR      -(SP)
          _TextFont
          CLR      -(SP)
          _TextFace
          MOVE     #12,-(SP)
          _TextSize
          CLR.L    -(SP)
          PEA     RectVTailles
          PEA     RectVTailles
          _TENew
          LEA     HandleTailles,A6
          MOVE.L   (SP)+,(A6)
          CLR     D4
          LEA     NombTailles,A3
          CLR     (A3)
          LEA     NumTailles,A4
          LEA     PoliceSelect,A6
          MOVE     (A6),D7
          ADD     D7,D7
          MOVE     0(A6,D7),D7
          LSR     #7,D7
          CLR.L   D6
          MOVEQ    #4,D6
          CLR.B   -(SP)
          MOVE     D7,-(SP)
          MOVE     D6,-(SP)
          _RealFont
          MOVE.B   (SP)+,D5
          BEQ     STaille
          ADDQ     #1,(A3)
          MOVE     D6,(A4)+
          MOVE.L   D6,D0
          LEA     TamponNombre,A0
          CLR     -(SP)
          _Pack7
          MOVE.B   (A0),D2
          MOVE.B   #32,(A0)
          MOVE.L   A0,-(SP)
          AND.L    #$FF,D2
          ADDQ     #1,D2
          MOVE.L   D2,-(SP)
          LEA     HandleTailles,A1
          MOVE.L   (A1),-(SP)
          _TEInsert
          PEA     CR
          MOVE.L   #2,-(SP)
          LEA     HandleTailles,A1
          MOVE.L   (A1),-(SP)
          _TEInsert
          ADDQ     #1,D6

```

```

          CMP     #128,D6
          BNE     BTaille
          MOVE     #1,2(A3)
          LEA     HandleTailles,A4
          MOVE.L   (A4),-(SP)
          _TEActivate
          LEA     NombTailles,A6
          CMP     #6,(A6)
          BLE.S   RetourT
          MOVE     (A6),D6
          SUBQ    #6,D6
          BRA.S   RetourT1
          CLR     D6
          LEA     HandleCil3,A4
          MOVE.L   (A4),-(SP)
          MOVE     D6,-(SP)
          _SelMaxCil
          MOVE.L   (A4),-(SP)
          CLR     -(SP)
          _SelCilValue
          LEA     HandleTailles,A6
          MOVEQ    #1,D7
          BSR     Selection
          MOVEM.L  (SP)+,A4
          RTS

```

```

RetourT1  LEA     HandleCil3,A4
          MOVE.L   (A4),-(SP)
          MOVE     D6,-(SP)
          _SelMaxCil
          MOVE.L   (A4),-(SP)
          CLR     -(SP)
          _SelCilValue
          LEA     HandleTailles,A6
          MOVEQ    #1,D7
          BSR     Selection
          MOVEM.L  (SP)+,A4
          RTS

```

```

; Traitement d'une action sur le bouton -
; 'Standard'

```

```

BSid      LEA     Apparence,A6
          MOVE     (A6),D0
          BEQ.S    BSidF
          CLR     (A6)
          LEA     HandleStand,A6
          MOVE.L   (A6),-(SP)
          MOVE     #1,-(SP)
          _SelCilValue
          MOVE     #4,D6
          CLR     D7
          LEA     HandleGras,A6
          MOVE.L   0(A6,D7),-(SP)
          CLR     -(SP)
          _SelCilValue
          ADDQ     #4,D7
          DBRA     D6,BSid
          BSidF   RTS

```

```

; Sous-programme pour caractéristiques du
; texte

```

```

CeracText LEA     PoliceSelect,A6
          MOVE     (A6),D7
          ADD     D7,D7
          MOVE     0(A6,D7),D7
          LSR     #7,D7
          MOVE     D7,-(SP)
          _TextFont
          LEA     TailleSelect,A6
          MOVE     (A6),D7
          ADD     D7,D7
          MOVE     0(A6,D7),-(SP)
          _TextSize
          LEA     Apparence,A6
          MOVE     (A6),-(SP)
          _TextFace
          PEA     FontInfo
          _SelFontInfo
          RTS

```

```

; Traitement d'une action sur le bouton "OK"

```

```

BOK       BSR     CeracTexte
          LEA     HandleTexte,A4
          MOVEA.L  (A4),A4
          MOVEA.L  (A4),A4
          LEA     FontInfo,A6
          MOVE     (A6),JeAscent(A4)
          MOVE     (A6),D0
          ADD     2(A6),D0
          ADD     8(A6),D0
          MOVE     D0,JeLineHite(A4)
          LEA     PoliceSelect,A6
          MOVE     (A6),D7
          ADD     D7,D7
          MOVE     0(A6,D7),D7
          LSR     #7,D7
          MOVE     D7,JeFont(A4)
          LEA     TailleSelect,A6
          MOVE     (A6),D7
          ADD     D7,D7
          MOVE     0(A6,D7),JeSize(A4)
          LEA     Apparence,A6
          MOVE.B   1(A6),JeFace(A4)
          LEA     RectDTexte,A6
          MOVE.L   (A6)+,(A4)+
          MOVE.L   (A6),A4
          LEA     HandleTexte,A4
          MOVE.L   (A4),-(SP)
          _TECalText
          PEA     RectVTexte
          _EraseRect
          PEA     RectVTexte
          LEA     HandleTexte,A6
          MOVE.L   (A6),-(SP)
          _TEUpdate
          LEA     HandleCil1,A8
          MOVE.L   (A8),-(SP)
          CLR     -(SP)
          _SelCilValue
          RTS

```

```

; Traitement d'une action sur la 'boisole'

```

```

DCoc      LEA     HandleTexte,A6
          MOVE.L   (A6),-(SP)
          _TEDispose
          PEA     RectVTexte

```

```

_EraseRect
CLR.L     -(SP)
PEA     RectDTexte
PEA     RectVTexte
_TENew
LEA     HandleTexte,A6
MOVE.L   (SP)+,(A6)
MOVE.L   (A6),-(SP)
_TEActivate
PEA     TamponPolice
MOVE.L   #446,-(SP)
MOVE.L   (A6),-(SP)
_TEInsert
LEA     HandleCil1,A6
MOVE.L   (A6),-(SP)
CLR     -(SP)
_SelCilValue
RTS

```

```

; Action dans la fenêtre de texte

```

```

DFTexte   LEA     EventRecord+10,A6
          MOVE     4(A6),D0
          ANDI     #$200,D0
          MOVE     #9,D1
          LSR     D1,D0
          MOVE.L   (A6),-(SP)
          MOVE.B   D0,-(SP)
          LEA     HandleTexte,A6
          MOVE.L   (A6),-(SP)
          _TEClick
          RTS

```

```

; Traitement d'une action sur un bouton
; permettant de modifier le style

```

```

BAppearance MOVE #1,D4
          SUBQ    #5,D1
          LSL     D1,D4
          LEA     Apparence,A6
          EOR     D4,(A6)
          MOVE     (A6),D6
          AND     D4,D6
          BEQ.S   StyleG2
          MOVE     #1,D4
          BRA.S   StyleG3
          CLR     D4
          LEA     WhichControl,A6
          MOVE.L   (A6),-(SP)
          MOVE     D4,-(SP)
          _SelCilValue
          LEA     Apparence,A6
          MOVE     (A6),D0
          BNE.S   StyleGS
          MOVE     #1,D4
          BRA.S   StyleGS2
          CLR     D4
          LEA     HandleStand,A6
          MOVE.L   (A6),-(SP)
          MOVE     D4,-(SP)
          _SelCilValue
          RTS

```

```

; Sélection d'une police

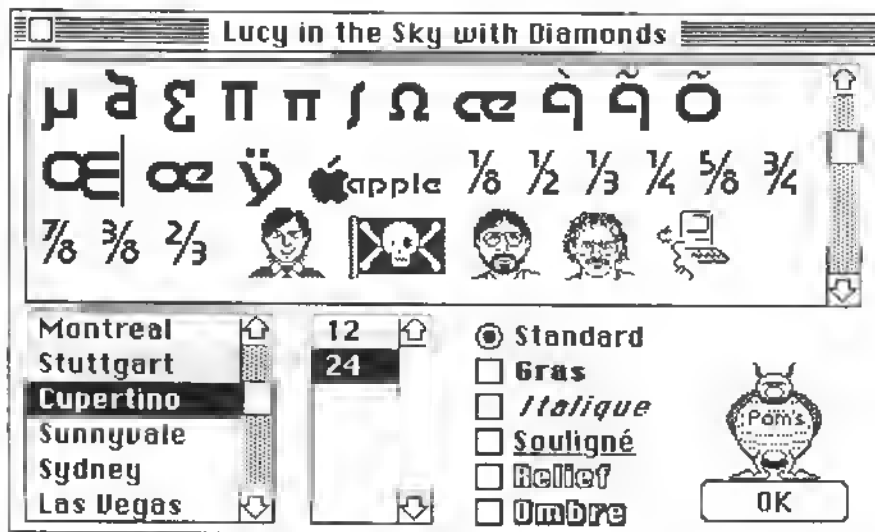
```

```

DFPolices PEA     PosMouse
          _SelMouse
          LEA     PosMouse,A6
          MOVE     (A6),D0
          CMPI     #128,D0
          BLE     DFPolices0
          CMPI     #144,D0
          BGT.S   DFPolices1
          MOVEQ    #1,D7
          BRA.S   DFPolicesT
          CMPI     #160,D0
          BGT.S   DFPolices2
          MOVEQ    #2,D7
          BRA.S   DFPolicesT
          CMPI     #176,D0
          BGT.S   DFPolices3
          MOVEQ    #3,D7
          BRA.S   DFPolicesT
          CMPI     #192,D0
          BGT.S   DFPolices4
          MOVEQ    #4,D7
          BRA.S   DFPolicesT
          CMPI     #208,D0
          BGT.S   DFPolices5

```





```

DFPolices5  MOVEQ    #5,D7
               BRA.S   DFPolicesT
               CMPI    #224,D0
               BGT.S   DFPolices0
               MOVEQ   #6,D7
DFPolicesT  LEA      NombPolices,A8
               MOVE    (A8),D6
               CMP     D6,D7
               BGT.S   DFPolices0
               LEA     HandleCtl2,A4
               CLR     -(SP)
               MOVE.L  (A4),-(SP)
               _GetCtlValue
               ADD     (SP)+,D7
               CMP     2(A6),D7
               BEQ.S   DFPolices0
               MOVE    D7,2(A6)
               LEA     HandlePolices,A8
               BSR     Selection
DFPolices0  BSR     TaillesP
               RTS

; Selection d'une taille

DFTailles  PEA      PosMouse
               _GetMouse
               LEA     PosMouse,A8
               MOVE    (A6),D0
               CMPI    #128,D0
               BLE     DFTailles0
               CMPI    #144,D0
               BGT.S   DFTailles1
               MOVEQ   #1,D7
DFPolices1  BRA.S   DFTaillesT
               CMPI    #190,D0
               BGT.S   DFTailles2
               MOVEQ   #2,D7
DFPolices2  BRA.S   DFTaillesT
               CMPI    #176,D0
               BGT.S   DFTailles3
               MOVEQ   #3,D7
DFPolices3  BRA.S   DFTaillesT
               CMPI    #192,D0
               BGT.S   DFTailles4
               MOVEQ   #4,D7
DFPolices4  BRA.S   DFTaillesT
               CMPI    #208,D0
               BGT.S   DFTailles5
               MOVEQ   #5,D7
DFPolices5  BRA.S   DFTaillesT
               CMPI    #224,D0
               BGT.S   DFTailles0
               MOVEQ   #6,D7
DFPolicesT  LEA      NombTailles,A8
               MOVE    (A6),D6
               CMP     D6,D7
               BGT     DFTailles0
               LEA     HandleCtl3,A4
               CLR     -(SP)
               MOVE.L  (A4),-(SP)
               _GetCtlValue
               ADD     (SP)+,D7
               CMP     2(A6),D7
               BEQ.S   DFTailles0
               MOVE    D7,2(A6)
               LEA     HandleTailles,A8
               BSR     Selection
DFTailles0  BSR     TaillesP
               RTS

; Fonctions standarda

Couper      LEA      HandleTailles,A8
               MOVE.L D7,2(A8)
               _TECut
               BRA     Finconrole

Couper0     BRA     Finconrole

Copier      LEA      HandleTailles,A8
               MOVE.L (A6),-(SP)
               _TECopy
               BRA.S   Couper0

Coller      LEA      HandleTailles,A8
               MOVE.L (A6),-(SP)
               _TEPaste
               BRA.S   Couper0

Effacer     LEA      HandleTailles,A8
               MOVE.L (A6),-(SP)
               _TEDelete
               BRA.S   Couper0

; Constantes et tampons

NombPolices DC 0
PoliceSelec DC 0
NumRefPolices DCB 40,0
NombTailles DC 0
TaillesSelec DC 0
NumTailles DCB 20,0
Apparence DC 0
ItemHit DC 0
ValeurCil DC 0
HautaurCarac DC 0
EvantRacord DCB 8,0
DialogPtr DC.L 0
HandleTexte DC.L 0
HandlePolices DC.L 0
HandleTailles DC.L 0
HandleCtl1 DC.L 0
HandleCtl2 DC.L 0
HandleCtl3 DC.L 0
HandleOK DC.L 0
HandleStand DC.L 0
HandleGras DC.L 0
HandleItal DC.L 0
HandleSoul DC.L 0
HandleRelief DC.L 0
HandleOmbre DC.L 0
HandleArt DC.L 0
WhichControl DC.L 0
PosMouse DC.L 0
HandleScrap DC.L 0
RectDial DC 50,100,281,490
RectTexte DC 7,7,119,377

```

```

RectPolices DC 127,7,224,104
RectTailles DC 127,135,224,176
RectDTexte DC 8,12,1000,364
RectVTexte DC 9,12,119,364
RectVPolices DC 129,8,223,103
RectVTailles DC 129,136,223,174
RectCoc DC 149,322,203,368
RectDrag DC 24,4,338,508
RectCtl1 DC 7,367,119,383
RectCtl2 DC 127,103,224,119
RectCtl3 DC 127,174,224,190
RectOK DC 203,310,222,377
RectStand DC 132,207,144,221
RectGras DC 148,207,160,221
RectItal DC 164,207,176,221
RectSoul DC 180,207,192,221
RectRelief DC 196,207,208,221
RectOmbre DC 212,207,224,221

```

```

FontInfo DCB 4,0
TamponNom DCB.B 32,32
TamponNombre DCB 2,0
BitMapCoc DC.L 0
Oct.LigneCoc DC 8
RectSrcCoc DC 0,0,54,44
CR DC.B 13,0
C1 DC.B 8,'Standard',0
C2 DC.B 4,'Gras',0
C3 DC.B 8,'Italique',0
C4 DC.B 8,'Souligné',0
C5 DC.B 8,'Relief',0
C6 DC.B 5,'Ombre'
TamponPolice DCB.B 446,0

```

```

BGETDITEM DCB 8,0
NomOK DCB 2,'OK',0
NomDefaut DCB 'Exemple de texte,'
           DCB 'Vous pouvez obtenir la'
           DCB ' police complète en cli'
           DCB 'quant sur la "bestiole".'

```

```

DataCoc DC.L $00080002,$00000004,$00040000
          DC.L $0006000C,$00000005,$000C0000
          DC.L $0008000C,$00000005,$3F940000
          DC.L $0002C088,$00000001,$00100000
          DC.L $00020008,$00000002,$7FC80000
          DC.L $0004B8A4,$00000004,$1F040000
          DC.L $00042084,$00000001E,$404F0000
          DC.L $006A40A4,$C000001DF,$209F7000
          DC.L $02B3DF79,$A8000661,$FFF0D400
          DC.L $0AD4FFE5,$6A001580,$3F803500
          DC.L $2B800000,$1A803600,$00000D80
          DC.L $6C1C1542,$06C05422,$00020540
          DC.L $A92273B0,$C2A0083C,$8A490360
          DC.L $A8208A48,$C2A0D020,$8A482160
          DC.L $B1207249,$C1A05400,$00000540
          DC.L $70600000,$01C03000,$00000180
          DC.L $31500001,$51801805,$55540300
          DC.L $08000000,$02000A00,$00000A00
          DC.L $04A80002,$A4000402,$AAAB80400
          DC.L $02000000,$0C000328,$00029800
          DC.L $01822A88,$300000C2,$40486000
          DC.L $00648024,$C000003C,$80278000
          DC.L $0018E0E3,$00000009,$FFF20000
          DC.L $0009FFF2,$00000004,$FFE40000
          DC.L $00049F24,$000000E2,$80C8E000
          DC.L $0317515D,$18001C08,$11020700
          DC.L $60002080,$00C00FFF,$C07FFF80

```

Lucy.R

Lucy.Rsrc

TYPE DRV = PROC
Lucy in the Sky with Diamonds ,28
Lucy

Lucy.Link

[

Lucy.Rel

\$

Fermez les fenêtres !

Julien Thomas



La version 2.0 du Basic Microsoft permet la gestion des fenêtres, donc leur fermeture. Cependant, trois fenêtres échappent à la règle : les deux fenêtres de "List" et la fenêtre de commande. Il est en effet impossible de fermer ces trois dernières depuis un programme Basic. Nous avons trouvé cela fâcheux car, si ces fenêtres ne sont pas fermées avant le début d'exécution du programme, l'utilisateur final peut les sélectionner, les déplacer et les fermer, ce qui a pour effet d'affecter la présentation du programme.

La solution consiste en l'emploi de la petite routine ici proposée, qui, placée en début de programme, efface toutes les fenêtres, une par une, jusqu'au "nettoyage" complet de l'écran. La méthode utilisée est simple : on appelle une routine du système qui retourne un pointeur sur la fenêtre située au premier plan, s'il n'y a pas de fenêtre (donc pas de pointeur), on revient au Basic, sinon, on appelle une seconde routine du système qui efface la fenêtre, et ainsi de suite...

Installation de la routine de suppression des fenêtres dans un programme Basic

```
DEFINT A-Z
DIM C(7)
' Data pour la routine
DATA &h42A7,&hA924,&h2E1F
DATA &h6706,&h2F07,&hA916
DATA &h60F2,&h4E75
' Implantation de la routine
FOR I=0 TO 7:READ C(I):NEXT I
' Appel de la routine (adresse dans A1)
A1=VARPTR(C(0)):A1
10 GOTO 10
```

Source de la routine d'effacement des fenêtres

0000	.Trap	_FrontWindow	\$A924	
0000	.Trap	_HideWindow	\$A916	
0000				
0000	42A7	Boucle	CLR.L	-(SP) ; Réserve 4 octets au sommet de la pile pour le pointeur
0002	A924		_FrontWindow	; Retourne un pointeur sur la fenêtre située au premier plan
0004	2E1F		MOVE.L	(SP)+,D7 ; Le résultat est dépilé et placé dans D7. Si le contenu de D7
0008	6706		BEQ.S	Retour ; est égal à 0, il n'y a plus de fenêtre à l'écran.
0008	2F07		MOVE.L	D7,-(SP) ; Place le pointeur sur la fenêtre au sommet de la pile
000A	A916		_HideWindow	; Efface la fenêtre
000C	60F2		BRA.S	Boucle ; Retour au début pour fenêtre suivante
000E	4E75	Retour	RTS	; Retour au Basic

Etrange accessoire !

Evelyne Gorin

Nous vous proposons un petit accessoire (que vous pourrez trouver sur la disquette Mac 19) qui, une fois lancé, ne peut plus être arrêté. En effet, le fait de passer d'une application au "Finder", ou du "Finder" à une application, ne perturbe absolument pas son fonctionnement. En dehors de cela, et mis à part son côté amusant, si d'aucun trouve un quelconque intérêt pratique à cette réalisation, nous vous serions reconnaissants de nous le faire savoir !

```
DC Ouv-Entr
DC Ouv-Entr
DC Ctr-Entr
DC Ouv-Entr
DC Ouv-Entr
```

```
STRING FORMAT 2
DC "5"
```

```
Buffer DC 0
```

```
Ouv MOVEO #0,D0
RTS
```

```
Ctr MOVEM.L A0-A2,-(SP)
LEA Buffer,A0
MOVEA.L #7A700,A1
MOVE #63,D3
_HideCursor
```

```
B0 MOVE #19,D0
```

```
CLR D1
```

```
B1 MOVE.B 0(A1,D1),(A0)
MOVEA.L A1,A2
```

```
B2 MOVEO #62,D2
MOVE.B 1(A2,D1),0(A2,D1)
ADDO.L #1,A2
DBRA D2,B2
MOVE.B (A0),63(A1,D1)
ADD #64,D1
DBRA D0,B1
DBRA D3,B0
_ShowCursor
MOVEM.L (SP)+,A0-A2
BRA.S Ouv
```

5'.Link

```
[
```

```
5'.Rel
```

```
$
```

5'.R

```
5'.Rsrc
```

```
TYPE DRV - PROC
```

```
5',29
```

```
5'
```

5'.Asm

```
.Trap _HideCursor $A852
.Trap _ShowCursor $A853
```

```
Entr DC $6400
DC 300
DC 0
DC 0
```



Le système de développement 68000

Cécile Derrien

Le système de développement 68000 est enfin disponible ! Comme nous l'attendions tous avec beaucoup d'impatience, une petite présentation s'impose.

Physiquement, le système est composé de deux disquettes, d'un manuel de référence du programmeur en 68000, du manuel du système de développement et, enfin, d'un câble destiné à relier le Mac à un second Mac ou à un Lisa. Pas de panique... le système fonctionne avec un seul Macintosh 128 ou 512Ko, avec ou sans lecteur externe (un second lecteur est toutefois recommandé, pour une raison de confort évidente). Le second Mac (ou Lisa) est seulement requis pour l'utilisation d'un "super debugger". Le système est livré en langue anglaise, et nous ne savons pas encore si une version française sera rapidement disponible.

Les disquettes

La première disquette (baptisée MDS1) contient, en plus du dossier système, l'éditeur, l'assembleur, le "linker", "l'executive" (difficile à traduire !), RMaker et PackSyms.

L'éditeur

Il s'agit d'un éditeur pleine page "Disk based" (donc pas de problèmes avec un Mac 128, les accès disquettes seront seulement plus fréquents qu'avec un 512) qui ne dépaysera personne puisque son mode de fonctionnement est semblable à celui de MacWrite; il supporte donc les désormais classiques "Copier, Couper, Coller", autorise le choix de la police de caractères ainsi que sa taille. Le choix du style (gras, italique, etc...) n'est pas permis, puisqu'il n'est d'aucun intérêt pour ce type d'application. La police non proportionnelle "Monaco", très peu employée en traitement de texte, se révèle être ici la plus lisible et, du même coup, la moins fatigante. Cet éditeur est déjà employé par "MacAdvantage" et nous aurons certainement l'occasion de le rencontrer souvent, puisqu'il est fourni pratiquement gratuitement par Apple aux développeurs de langages informatiques qui en font la demande.

L'assembleur

Cet assembleur est à la hauteur des possibilités du MC 68000 : directives nombreuses, établissement de macro-instructions aisé, etc... et même comportement "intelligent" : si vous placez, dans la source, le code mnémotique "ADD.L #4,D0", qui produit le code machine \$D1BC \$0000 \$0004 (soit trois mots ou encore six octets), il sera automatiquement remplacé par "ADDQ.L #4,D0", qui produit le code \$5880. Le résultat est le même, mais avec cette fois seulement deux octets.

Un seul reproche à faire à cet assembleur : il ne produit pas un code objet directement exécutable; pour cela, il faut absolument passer par le "Linker". Si cela n'a pas d'importance

pour une utilisation normale (écriture d'une application) il n'en va pas de même lorsque l'on écrit une routine destinée à être utilisée depuis un programme Basic car, dans ce cas, il est nécessaire de calculer soi-même certains branchements (l'assembleur ne calcule pas les branchements "vers le bas").

Le "linker"

Il transforme le code objet relogé produit par l'assembleur en une application. Si besoin est, il se charge aussi de la connection de modules assemblés séparément. C'est aussi lui qui autorise l'établissement des "signatures"; par exemple : APPL TEXT (voir l'article "Modification des informations pour le Finder", dans le précédent numéro de Pom's).

RMaker

Le Macintosh utilise des "ressources" pour la définition des fenêtres, les contrôles, les menus, les chaînes de caractères, les icônes, etc... En somme, tout ce qui n'est pas du code exécutable par le processeur (bien que l'on puisse aussi trouver du code 68000 dans un "fichier ressource"). Ces définitions peuvent aussi être incluses dans le source; c'est la méthode employée par l'accessoire "Lucy". Le principe des ressources est intéressant pour deux raisons : si un programme doit être commercialisé dans plusieurs pays, les messages et menus peuvent être adaptés (avec un éditeur de ressource) à la langue locale, et ce sans qu'il soit nécessaire de modifier le code objet. De plus l'établissement de ressources est très simple puisque l'on met à notre disposition un "compilateur de ressources". Nous avons cité "RMaker".

Par exemple, pour créer une ressource destinée à la définition d'un menu, il suffit d'établir avec l'éditeur une liste qui pourrait ressembler à :

TYPE MENU

```
,1
Menu général
Début
Bof
Fin
```

1 correspond à l'identificateur du menu, "Menu général" correspond au titre (qui apparaît dans la barre des menus), le reste se rapporte aux trois articles du menu. Simple non ?

Executive

"Executive" est un utilitaire extrêmement pratique bien que très simple. Lorsque l'on écrit une application, il faut passer par plusieurs étapes : édition, assemblage, "linkage", compilation des ressources. Executive autorise une exécution automatique du processus, ce qui permet un gain de temps appréciable. De plus, l'enchaînement des tâches peut être conditionnel : par exemple, on peut décider qu'après l'assemblage, le "linker" sera appelé en cas de succès (assemblage sans erreur, de syntaxe ou autre), alors que la main sera passée à l'éditeur dans le cas contraire.

A noter qu'il est aussi possible d'appeler une application depuis une autre par l'intermédiaire du menu "Transfert", qui agit un peu comme le "Switcher" pour Mac 512Ko.

PackSyms

Cet utilitaire permet d'obtenir des fichiers de symboles (TOTO EQU \$23, GAGA EQU TOTO+4, etc...) sous une forme compactée. Ceci permet de gagner de la place en mémoire, sur les disquettes, et aussi d'accélérer les traitements.

La seconde disquette (MDS2) contient des fichiers de symboles, des exemples de programmes en assembleur 68000, et enfin les "Debuggers".

Fichiers de symboles

Les fichiers de symboles sont prévus pour être utilisés depuis un source (avec la directive INCLUDE) et contiennent les "Traps", les adresses du système, les décalages par rapport à la base des différents tableaux ou tampons, les équivalences pour les codes d'erreur générés par le système d'exploitation, etc... L'utilisation de ces fichiers est fortement conseillée car, outre le gain de temps apporté, elle permet une amélioration de la lisibilité des sources, ce qui n'est pas désagréable.

Les exemples

Le système est fourni avec quelques exemples propres à faciliter les premiers pas du développeur en assembleur 68000. Parmi ceux-ci, nous trouvons :

- gestion d'une fenêtre;
- impression depuis une application;
- un petit programme utilisant quelques routines "QuickDraw";
- un accessoire de bureau "minimum" (il affiche simplement une fenêtre).

Ces exemples sont quelque peu limités, mais permettent néanmoins de débiter. Pour aller plus loin, lisez Pom's !

Les "debuggers"

Trois programmes de mise au point sont à notre disposition. Le premier, baptisé MacsBug, fonctionne sur un Mac 128Ko. Le second (Maxbug) est prévu pour un 512Ko. Le troisième (MacDB) est le plus puissant (certainement le plus puissant jamais rencontré sur un micro-ordinateur) mais requiert l'utilisation d'un second Mac ou d'un Lisa.

Inside Macintosh

Nous réclamions une documentation technique depuis la sortie du Macintosh. Nous ne savons pas si le fait d'avoir "ralé" dans de précédents articles y est pour quelque chose, mais elle arrive ! Inside Macintosh est un ouvrage d'environ 1800 pages indispensable à qui veut programmer sérieusement le Mac.

Création de caractères

Jean-Pierre Vinter

Avec ce nouvel utilitaire, les possesseurs d'un Apple II muni d'une imprimante de la série Imagewriter, vont enfin pouvoir créer et exploiter commodément tous les caractères qu'ils désirent. S'ils possèdent, en outre, la disquette "Basic Purplesoft" distribuée avec la carte "Chat Mauve" ils vont, de plus, disposer de police de caractères très diverses (Italiques, Gothiques ou Grecques), directement disponibles et exploitables par l'imprimante.

Ce programme comble une lacune du "Tool kit Imagewriter" qui ne permet que l'utilisation de caractères courants réduits et décalés vers le haut, ou vers le bas.

Il est composé d'un programme de lancement et de gestion écrit en Basic qui fait largement appel à différents niveaux d'une routine assembleur dont les fonctions sont :

- la création
- la modification
- l'ajout
- la suppression des caractères en mémoire.

Le programme Basic pouvant, dès lors, les envoyer sur imprimante ou les sauvegarder pour une future utilisation.

Codage et décodage des caractères

A l'origine du programme Basic, se posait le problème suivant : comment récupérer facilement un jeu de caractères graphiques existant sur les disquettes "Chat Mauve" et le rendre exploitable par "Imagewriter".

Bien que le principe soit semblable, le codage d'un caractère se fait différemment selon qu'il est utilisé par la haute résolution graphique ou par l'interface imprimante. Ainsi, à titre d'exemple, si l'on veut coder un "A", sur une largeur de 8 dots, il se représentera par la matrice binaire désignée ci-dessous :

			●	●	●		
		●	●		●	●	
		●				●	
	●	●				●	●
	●	●	●	●	●	●	●
	●	●				●	●
	●	●				●	●

0	0	0	1	1	1	0	0	38
0	0	1	1	0	1	1	0	6C
0	0	1	0	0	0	1	0	44
0	1	1	0	0	0	1	1	C6
0	1	1	1	1	1	1	1	FE
0	1	1	0	0	0	1	1	C6
0	1	1	0	0	0	1	1	C6
0	0	0	0	0	0	0	0	00
00	7D	78	13	11	13	7D	78	

Cette matrice binaire sera codée en hexadécimal :

- de droite à gauche et de bas en haut s'il s'agit de la haute résolution graphique : 00 C6 C6 FE C6 44 6C 38
- de bas en haut et de gauche à droite s'il s'agit de l'interface imprimante soit : 00 78 7D 13 11 13 77 78

En outre, dans le cas de l'imprimante, les 8 octets ainsi codés devront être précédés de 2 codes :

- le premier caractérisant la touche concernée du clavier, ici un "A"
- le second représentant sa longueur, ici la représentation du chiffre "8".

Le renversement de la matrice, nécessaire pour la traduction "Chat Mauve - Imagewriter", ne pouvait se faire qu'en langage machine. En effet, la solution Basic qui consistait à lire les octets, les convertir en binaire, les charger dans un tableau que l'on lisait différemment avant de les recoder, était trop gourmande en temps, près de 10 minutes pour 100 caractères.

Pour être complet sur ce sujet, rappelons également qu'un caractère "Imagewriter" est en fait constitué par une matrice de 9 aiguilles de haut, numérotées de 1 à 9 et que la 9ème aiguille est utilisée pour le soulignement. L'utilisateur peut indifféremment choisir les rangées 1 à 8, ou 2 à 9, selon qu'il code la largeur des caractères par une lettre majuscule ou minuscule. La largeur de la matrice est de 16 aiguilles au maximum, les codes correspondants sont, par convention :

- "A" pour 1, "B" pour 2, ..., "P" pour 16 (dots hauts de 1 à 8)
- "a" pour 1, "b" pour 2, ..., "p" pour 16 (dots bas de 2 à 9)

L'alphabet standard a une largeur constante de 8 dots, il utilise les ran-

gées 1 à 7 pour les lettres majuscules et 3 à 7 pour les minuscules, les rangées 8 et/ou 9 étant réservées aux lettres qui se prolongent vers le bas (ç, g, j, p, q et y).

Chargement de l'imprimante

Un problème a dû être résolu :

Il était nécessaire d'utiliser un langage d'assemblage pour envoyer les codes à l'imprimante. En effet, l'Applesoft positionne le bit de poids fort des octets à 1 et un PRINT "Caractère" ou par un "PRINT CHR\$(code décimal)", pour la création de caractères, entraîne un soulignement gênant, les dots 8 ou 9 de la matrice étant parasités par ces PRINTs. Mais il faut également que l'imprimante reconnaisse tous les bits qui vont lui parvenir. De ce fait, pour être utilisable, chaque fichier devra commencer et finir par les séquences d'ESCAPE suivantes :

- au début : ESC Z CONTROLà (1B 44 00 20 en hexa), pour reconnaître le 8ème bit
- à la fin : ESC D CONTROLà (1B 5A 00 20 en hexa), pour ignorer le 8ème bit, qui est l'utilisation normale de l'imprimante en mode texte.

Le caractère "-" représente l'espace.

Présentation du programme Basic

Place mémoire

L'instruction HIMEM: 16383 réserve la mémoire à partir de l'adresse hexadécimale \$4000. C'est à partir de cette adresse que sera stockée la police de caractères à créer ou à traiter. Si une police est composée de 95 caractères (codes ASCII de 32 à 127), tous directement adressables par le clavier et ayant chacun une largeur maximale de 16 dots, l'occupation mémoire serait, au plus, un peu supérieure à 1,5 Ko (16+2 octets * 95).

La routine en langage machine est implantée à partir de l'adresse \$6000 et occupe 858 octets. elle ne peut pas être parasitée par la mémoire caractère.

Menu

Le menu principal offre les possibilités suivantes :

- 1- Créer un nouveau fichier;
- 2- Lui ajouter de nouveaux caractères;

- 3- Afficher ou modifier les caractères d'un fichier;
- 4- Récupérer un jeu de caractères
- 5- Sauvegarder et/ou charger un fichier sur imprimante.

L'option 4 est spécifique aux fichiers "Chat Mauve". Elle assure la récupération et la conversion des 95 caractères directement adressables (codes ASCII 32 à 127). Pour éviter les confusions de fichiers, le programme ajoute lui-même les suffixes ".FONT" relatifs aux polices de caractères "Chat Mauve" et ".CHAR" pour les fichiers qu'il a lui-même créés. L'utilisateur se contentera d'indiquer la première partie du nom du fichier, par exemple "GOTHIC" pour des caractères gothiques.

Options

La différence essentielle entre les options 2 et 3, ajout et modification de données, tient dans le fait que, dans l'option 2, le programme ajoute une nouvelle forme à la fin du fichier, alors que dans l'option 3, si une forme est définie avec un code caractère déjà existant, celle-ci sera, après confirmation, éliminée physiquement du fichier, puis remplacée par la nouvelle forme.

L'option 2 sera donc, en principe, choisie par un utilisateur qui n'aurait pas fini de rentrer tous les caractères qu'il s'était proposé de créer. En fin de création (option 1) ou d'ajouts de formes à un fichier (option 2), une réponse NON à la question "Nouvelle forme?" renvoie à l'option 3, examen / modification qui seule permet l'accès au sous-menu, sauvegarde / impression de fichier (option 5).

L'option 3 appelle un caractère existant dans le fichier, que ce soit pour le consulter ou pour le modifier. Un menu permet de le supprimer du fichier, de le modifier ou de passer à un autre caractère. Notons que RETURN en réponse à la demande du code du caractère, permet de faire apparaître l'enregistrement qui suit celui qui vient d'être appelé ou le premier caractère du fichier (si aucun n'avait encore été appelé ou si une suppression venait d'être effectuée). Si une modification a été faite (sous-option 2), il est possible de mémoriser la nouvelle forme, soit en conservant son code, soit sous un nom de code différent. Remarquons qu'il est aussi possible de créer de nouveaux caractères avec l'option 3. Il suffit d'appeler un caractère déjà existant dans le fichier, d'éventuellement le modifier et de le stocker en mémoire sous un nom de code encore inexistant. Ainsi, à titre d'exemple :

— une utilisation judicieuse de l'option 3 pour des français serait, après récupération d'un fichier "Chat

Mauve" non accentué, d'appeler le "e", de lui ajouter un accent et de le stocker sous un nom de code différent, "é" ou "è",...

— De même, si l'on veut créer une frise décorative, on appellera et on modifiera un caractère qui ne sera pas utilisé dans le texte, caractère que l'on stockera, dans ce cas, avec le même nom de code.

Gestion d'écran

Qu'il s'agisse de création ou de modification de formes, l'affichage est matérialisé sur l'écran par un cadre de 8 cases de haut sur 16 de large, bordé en haut et en bas par les lettres A à P, majuscules et minuscules, pour rappeler la convention du code largeur. Le curseur se meut librement à l'intérieur du cadre au moyen des 4 flèches disponibles sur l'Apple II/e (pour l'Apple II+, on utilisera les flèches gauche et droite et les habituels CTRL-J et CTRL-K). Une pression sur la barre d'espacement permet de mettre à 1 ou à 0 le bit de représentation d'un dot du caractère, cette action étant visualisée par une étoile ou un espace sur l'écran, qui assure une image très agrandie de ce que sera le caractère imprimé. Une option supplémentaire, qui n'apparaît pas à l'écran, est accessible au moyen de la touche "?". Elle permet des décalages de la forme affichée sur l'écran, selon les 4 directions habituelles (haut, bas, droite et gauche).

Une pression sur la touche "0" permet de sortir du mode création — modification. Il sera alors demandé successivement les codes caractère et largeur de la forme. La largeur détermine le nombre de colonnes qui seront interprétées, à partir de la colonne 1. Elle est laissée au libre choix de l'utilisateur qui peut donc ne mémoriser qu'une partie de l'écran. La touche RETURN en réponse à une des deux questions, code ou largeur, renvoie l'affichage du caractère.

Sauvegarde et chargement

L'envoi vers l'imprimante nécessite que celle-ci soit raccordée et mise sous tension avant l'envoi des informations. Un test optionnel permet d'imprimer les caractères dans les 2 modes, standard et police de caractères étrangère. La commutation des 2 modes se faisant par la séquence d'ESCAPE suivante :

- caractères créés : ESC ' (1B 27 en hexadécimal)
- caractères standards : ESC \$ (1B 24 en hexadécimal)

La sauvegarde ne nécessite que la réponse à la question "Nom du fichier". Aucun contrôle n'étant effectué au niveau du DOS, il conviendra de faire attention à ne pas donner le nom d'un fichier déjà existant sous

peine de son remplacement sur la disquette. Cette option représente la dernière opération à effectuer sur une police de caractères. Après exécution, l'utilisateur est renvoyé au menu principal, dont l'option 0 constitue la sortie normale du programme. Il pourra aussi, traiter ou créer d'autres fichiers ou rappeler un précédent fichier de travail, si celui-ci a été sauvegardé.

Annexes

Les listings des programmes, Basic et Assembleur sont présentés ci-après. Pour ceux qui ne disposeraient pas de carte Chat Mauve, figure également ci-après un jeu de caractères créé à leur attention. Il s'agit d'une police de caractères français, accentués, inspirée des caractères de type BYTE. Ils devront être chargés en mémoire à partir de l'adresse \$4000, la longueur du fichier est de 865 octets (\$361 en hexadécimal). Une fois le fichier sauvegardé sous un nom choisi, il pourra être envoyé vers Imagewriter au moyen de l'option 5 du programme. La séquence d'ESCAPE de fin de fichier n'a pas à figurer dans le fichier, elle est assurée par cette option du programme. La routine en Assembleur a été conçue de façon très modulaire et ne nécessite pas d'explications particulières. Le listing du programme source comporte pour chaque module :

- un titre en commentaire qui rappelle sa fonction;
- un label qui est le même que celui qui figure en tête du programme Basic.

N.D.L.R. : Afin que l'impression des polices de caractères fonctionne sur l'imprimante DMP, il faut :

- envoyer, en début de programme, le code de neutralisation de l'interface
- rajouter en ligne 121 du programme Basic les POKEs nécessaires, pour obtenir dans le code objet de SAISIE :

618A : 20 ED FD 60

Programme

SAISIE.S (Assembleur MERLIN)

```

1 *****
2 * PRGRM DE TRAITEMENT DE
3 * CARACTERES IMAGEWRITER
4 *****
5          ORG    $6000
6 ZERO    =    $00
7 UN      =    $01
8 DEP     =    $06
9 SIX     =    $06

```

10	SEPT	=	\$07	75	TYA	140	ADC	H			
11	HUIT	=	\$08	76	ADC	ARR	141	STA	ADB		
12	ARR	=	\$08	77	STA	\$4000	142	LDY	#ZERO		
13	NEUF	=	\$09	78	LDA	NEUF	143	LDA	{ADB},Y		
14	DIX	=	\$10	79	ADC	#ZERO	144	CMP	#STAR		
15	LON	=	\$18	80	STA	\$4001	145	BNE	P43		
16	ASCI	=	\$18	81	RTS		146	INC	\$1A		
17	CARG	=	\$19	82	*****		147	P43	DEC	V	
18	H	=	\$24	83	* FIN CHARGEMENT FORME		148	DEX			
19	V	=	\$25	84	* ESCAP FIN FICHER/ADRESS		149	CPX	#ZERO		
20	ESC	=	\$1B	85	* ENVOI VERS PRINTER		150	BNE	P42		
21	BLAN	=	\$A0	86	*****		151	*****			
22	STAR	=	\$AA	87	P30	LDY	#ZERO	152	INC	H	
23	ADB	=	\$28	88		LDA	ARR	153	INC	LON	
24	CALC	=	\$300	89		STA	\$4000	154	LDA	LON	
25	COMPAR	=	\$308	90		LDA	NEUF	155	CMP	#\$10	
26	REGIS	=	\$309	91		STA	\$4001	156	BEQ	P44	
27	VTAB	=	\$FC22	92		LDA	#04	157	TAX		
28	BASCAL	=	\$FBC1	93		STA	{ARR},Y	158	LDA	\$1A	
29	*****			94		INY		159	STA	CALC,X	
30	P01	JSR	VTAB	95		LDA	#ESC	160	JMP	P41	
31		LDA	V	96		STA	{ARR},Y	161	P44	RTS	
32		JSR	BASCAL	97		INY		162	*****		
33		LDA	ADB	98		LDA	#\$44	163	* PROGRAMME	CHAT MAUVE	
34		CLC		99		STA	{ARR},Y	164	* CHARGE	ADRESSE ET CONST	
35		ADC	H	100		INY		165	*****		
36		STA	ADB	101		LDA	#ZERO	166	P50	LDA	#ZERO
37		LDY	#ZERO	102		STA	{ARR},Y	167		STA	DEP
38	*****			103		INY		168		LDA	#\$41
39		LDA	{ADB},Y	104		LDA	#\$20	169		STA	SEPT
40		CMP	#STAR	105		STA	{ARR},Y	170		LDA	#\$1F
41		BNE	P02	106	*****			171		STA	ASCI
42		LDA	#BLAN	107		CLC		172		LDA	#\$47
43		STA	{ADB},Y	108		LDA	ARR	173		STA	CARG
44		RTS		109		ADC	#\$05	174		JSR	SP10
45	*****			110		STA	ARR	175	*****		
46	P02	LDA	#STAR	111		LDA	NEUF	176	P51	LDY	#ZERO
47		STA	{ADB},Y	112		ADC	#ZERO	177	P52	LDA	{DEP},Y
48		RTS		113		STA	NEUF	178		STA	CALC,Y
49	*****			114		LDA	#\$02	179		INY	
50	*	FIN	CONVERSION	115		STA	DEP	180		CPY	#HUIT
51	*****			116		LDA	#\$40	181		BNE	P52
52	* CHARGEMENT	ESCAPE	IMPRIM	117		STA	SEPT	182	*****		
53	P10	JSR	SP10	118		JMP	SP1	183		LDA	#\$80
54	*****			119	*****			184		STA	COMPAR
55	P20	LDY	#ZERO	120	* LIT	ECRAN ET CHARGEMENT		185		LDA	#ZERO
56		LDX	#ZERO	121	*	EN	\$300	186		STA	REGIS
57		LDA	\$06	122	*****			187		LDX	#ZERO
58		STA	{ARR},Y	123	P40	LDA	#\$FF	188	P53	LSR	COMPAR
59		INY		124		STA	LON	189		LDY	#ZERO
60		LDA	SEPT	125		LDA	#\$0B	190	P54	ASL	REGIS
61		STA	{ARR},Y	126		STA	H	191		LDA	\$300,Y
62		INY		127	*****			192		AND	COMPAR
63	*****			128	P41	LDA	#ZERO	193		BEQ	P55
64	P21	LDA	CALC,X	129		STA	\$1A	194		INC	REGIS
65		STA	{ARR},Y	130		LDY	#ZERO	195	P55	INY	
66		INX		131		LDX	#HUIT	196		CPY	#HUIT
67		INY		132		LDA	#\$10	197		BNE	P54
68		DEC	LON	133		STA	V	198		LDA	REGIS
69		LDA	LON	134	*****			199		STA	\$310,X
70		CMP	#ZERO	135	P42	ASL	\$1A	200		INX	
71		BEQ	P22	136		LDA	V	201		CPX	#SEPT
72		JMP	P21	137		JSR	BASCAL	202		BNE	P53
73	*****			138		CLC		203	*****		
74	P22	CLC		139		LDA	ADB	204		LDY	#ZERO

205	INC	ASCI	270	LDY	#DIX	335	INX
206	LDA	ASCI	271	*****		336	LDA LON
207	STA	(ARR),Y	272	SP5	LDA \$C089,Y	337	CMP #\$60
208	INY		273	AND	#DIX	338	BEQ P63
209	LDA	CARG	274	BEQ	SP5	339	CMP #\$40
210	STA	(ARR),Y	275	LDA	\$C089	340	BNE P62
211	LDX	#SIX	276	AND	#\$40	341	*****
212	INY		277	BNE	SP5	342	P63 CLC
213	*****		278	TXA		343	LDY #ZERO
214	P56	LDA \$310,X	279	STA	\$C088,Y	344	TXA
215	STA	(ARR),Y	280	STA	\$C080,Y	345	ADC DEP
216	INY		281	RTS		346	STA DEP
217	DEX		282	*****		347	LDA SEPT
218	CPY	#NEUF	283	* ESC DEBUT FICHIER IMPRI		348	ADC #ZERO
219	BNE	P56	284	*****		349	STA SEPT
220	*****		285	SP10	LDY #ZERO	350	CMP \$4001
221	LDA	DEP	286	LDA	#\$02	351	BNE P61
222	CLC		287	STA	ARR	352	LDA DEP
223	ADC	#HUIT	288	LDA	#\$40	353	CMP \$4000
224	STA	DEP	289	STA	NEUF	354	BNE P61
225	BCC	P57	290	LDA	#ESC	355	LDA #FFF
226	INC	SEPT	291	STA	(ARR),Y	356	STA HUIT
227	*****		292	INC	ARR	357	P64 RTS
228	P57	LDA ARR	293	LDA	#\$5A	358	*****
229	CLC		294	STA	(ARR),Y	359	* CHARGEMENT EN \$300
230	ADC	#NEUF	295	INC	ARR	360	*****
231	STA	ARR	296	LDA	#ZERO	361	P65 LDY #\$01
232	BCC	P58	297	STA	(ARR),Y	362	LDA (DEP),Y
233	INC	NEUF	298	INC	ARR	363	STA LON
234	*****		299	LDA	#\$20	364	LDX #ZERO
235	P58	LDA ASCI	300	STA	(ARR),Y	365	INY
236	CMP	#\$7E	301	INC	ARR	366	P66 LDA (DEP),Y
237	BNE	P51	302	LDA	#ESC	367	STA CALC,X
238	LDA	#\$61	303	STA	(ARR),Y	368	INX
239	STA	\$4000	304	INC	ARR	369	INY
240	LDA	#\$43	305	LDA	#\$2B	370	DEC LON
241	STA	\$4001	306	STA	(ARR),Y	371	LDA LON
242	RTS		307	INC	ARR	372	CMP #\$60
243	*****		308	LDA	#ESC	373	BEQ P67
244	* FIN PGM ~ ROUTINE CHARGT		309	STA	(ARR),Y	374	CMP #\$40
245	*****		310	INC	ARR	375	BNE P66
246	SP1	LDA DEP	311	LDA	#\$49	376	*****
247	CMP	ARR	312	STA	(ARR),Y	377	P67 TXA
248	BNE	SP2	313	INC	ARR	378	CMP #\$10
249	LDA	SEPT	314	RTS		379	BEQ P68
250	CMP	NEUF	315	*****		380	LDA #ZERO
251	BEQ	SP3	316	* RECHERCHE D'UN CARACTERE		381	STA CALC,X
252	*****		317	*****		382	INX
253	SP2	LDY #ZERO	318	P60	LDA #\$0A	383	JMP P67
254	LDA	(DEP),Y	319	STA	DEP	384	P68 RTS
255	JSR	SP4	320	LDA	#\$40	385	*****
256	CLC		321	STA	SEPT	386	* AFFICHAGE CARACTERE
257	LDA	DEP	322	LDA	HUIT	387	*****
258	ADC	#UN	323	STA	CARG	388	P70 LDX #ZERO
259	STA	DEP	324	LDY	#ZERO	389	LDA #0B
260	LDA	SEPT	325	*****		390	STA H
261	ADC	#ZERO	326	P61	LDA (DEP),Y	391	*****
262	STA	SEPT	327	CMP	CARG	392	P71 LDA #\$80
263	SEC		328	BEQ	P64	393	STA LON
264	BCS	SP1	329	INY		394	LDA CALC,X
265	SP3	RTS	330	LDA	(DEP),Y	395	STA CARG
266	*****		331	STA	LON	396	LDY #ZERO
267	* BOUCLES D'ATTENTE		332	LDX	#\$02	397	LDA #\$10
268	*****		333	*****		398	STA V
269	SP4	TAX	334	P62	DEC LON	399	*****

400 P72	LDA V	447	LDY #ZERO	494	RTS
401	JSR BASCAL	448 *****		495 *****	
402	LDA ADB	449	LDA ARR	496 *	DECALAGE BAS
403	CLC	450	CMP \$4000	497 *****	
404	ADC H	451	BNE P83	498	LDX #ZERO
405	STA ADB	452	LDA NEUF	499 S20	LDA CALC,X
406	LDA CARG	453	CMP \$4001	500	ASL
407	AND LON	454	BNE P83	501	STA CALC,X
408	BEQ P73	455	JMP P84	502	INX
409	LDA #STAR	456 *****		503	CPX #\$10
410	JMP P74	457 P83	LDA (ARR),Y	504	BNE S20
411 P73	LDA #BLAN	458	STA (DEP),Y	505	RTS
412 P74	STA (ADB),Y	459	CLC	506 *****	
413	DEC V	460	LDA DEP	507 *	DECALAGE DROITE
414	LSR LON	461	ADC #\$01	508 *****	
415	LDA LON	462	STA DEP	509	LDX #\$0E
416	CMP #ZERO	463	LDA SEPT	510 S30	LDA CALC,X
417	BNE P72	464	ADC #ZERO	511	INX
418 *****		465	STA SEPT	512	STA CALC,X
419	INC H	466	CLC	513	DEX
420	INX	467	LDA ARR	514	DEX
421	TXA	468	ADC #UN	515	CPX #ZERO
422	CMP #\$10	469	STA ARR	516	BNE S30
423	BNE P71	470	LDA NEUF	517	LDA CALC,X
424	RTS	471	ADC #ZERO	518	INX
425 *****		472	STA NEUF	519	STA CALC,X
426 * SUPPRESSION D'UNE FORME		473	CMP \$4001	520	LDA #ZERO
427 *****		474	BNE P83	521	STA CALC
428 P80	LDY #\$01	475	LDA ARR	522	RTS
429	LDA (DEP),Y	476	CMP \$4000	523 *****	
430	STA LON	477	BNE P83	524 *	DECALAGE GAUCHE
431	LDX #S02	478 *****		525 *****	
432 P81	DEC LON	479 P84	LDA DEP	526	LDX #\$01
433	INX	480	STA \$4000	527 S40	LDA CALC,X
434	LDA LON	481	LDA SEPT	528	DEX
435	CMP #\$60	482	STA \$4001	529	STA CALC,X
436	BEQ P82	483	RTS	530	INX
437	CMP #\$40	484 *****		531	INX
438	BNE P81	485 *	DECALAGE HAUT	532	CPX #\$0F
439 *****		486 *****		533	BNE S40
440 P82	CLC	487	LDX #ZERO	534	LDA CALC,X
441	TXA	488 S10	LDA CALC,X	535	DEX
442	ADC DEP	489	LSR	536	STA CALC,X
443	STA ARR	490	STA CALC,X	537	LDA #ZERO
444	LDA SEPT	491	INX	538	STA \$30F
445	ADC #ZERO	492	CPX #\$10	539	RTS
446	STA NEUF	493	BNE S10	540 *****	

Récapitulation SAISIE.OBJ

6000- 20 22 FC A5 25 20 C1 FB
6008- A5 28 18 65 24 85 28 A0
6010- 00 B1 28 C9 AA D0 05 A9
6018- A0 91 28 60 A9 AA 91 28
6020- 60 20 A3 61 A0 00 A2 00
6028- A5 06 91 08 C8 A5 07 91
6030- 08 C8 BD 00 03 91 08 E8
6038- C8 C6 18 A5 18 C9 00 F0
6040- 03 4C 32 60 18 98 65 08
6048- 8D 00 40 A5 09 69 00 8D
6050- 01 40 60 A0 00 A5 08 8D

6058- 00 40 A5 09 8D 01 40 A9
6060- 04 91 08 C8 A9 1B 91 08
6068- C8 A9 44 91 08 C8 A9 00
6070- 91 08 C8 A9 20 91 08 18
6078- A5 08 69 05 85 08 A5 09
6080- 69 00 85 09 A9 02 85 06
6088- A9 40 85 07 4C 66 61 A9
6090- FF 85 18 A9 0B 85 24 A9
6098- 00 85 1A A0 00 A2 08 A9
60A0- 10 85 25 06 1A A5 25 20
60A8- C1 FB 18 A5 28 65 24 85
60B0- 28 A0 00 B1 28 C9 AA D0
60B8- 02 E6 1A C6 25 CA E0 00
60C0- D0 E1 E6 24 F6 18 A5 18
60C8- C9 10 F0 09 AA A5 1A 9D

60D0- 00 03 4C 97 60 60 A9 00
60D8- 85 06 A9 41 85 07 A9 1F
60E0- 85 18 A9 47 85 19 20 A3
60E8- 61 A0 00 B1 06 99 00 03
60F0- C8 C0 08 D0 F6 A9 80 8D
60F8- 08 03 A9 00 8D 09 03 A2
6100- 00 4E 08 03 A0 00 0E 09
6108- 03 B9 00 03 2D 08 03 F0
6110- 03 EE 09 03 C8 C0 08 D0
6118- ED AD 09 03 9D 10 03 E8
6120- E0 07 D0 DD A0 00 E6 18
6128- A5 18 91 08 C8 A5 19 91
6130- 08 A2 06 C8 BD 10 03 91
6138- 08 C8 CA C0 09 D0 F5 A5
6140- 06 18 69 08 85 06 90 02

```

6148- E6 07 A5 08 18 69 09 85
6150- 08 90 02 E6 09 A5 18 C9
6158- 7E D0 8E A9 61 8D 00 40
6160- A9 43 8D 01 40 60 A5 06
6168- C5 08 D0 06 A5 07 C5 09
6170- F0 17 A0 00 B1 06 20 8A
6178- 61 18 A5 06 69 01 85 06
6180- A5 07 69 00 85 07 38 B0
6188- DD 60 AA A0 10 B9 89 C0
6190- 29 10 F0 F9 AD 89 C0 29
6198- 40 D0 F2 8A 99 88 C0 99
61A0- 80 C0 60 A0 00 A9 02 85
61A8- 08 A9 40 85 09 A9 1B 91
61B0- 08 E6 08 A9 5A 91 08 E6
61B8- 08 A9 00 91 08 E6 08 A9
61C0- 20 91 08 E6 08 A9 1B 91
61C8- 08 E6 08 A9 2B 91 08 E6
61D0- 08 A9 1B 91 08 E6 08 A9
61D8- 49 91 08 E6 08 60 A9 0A
61E0- 85 06 A9 40 85 07 A5 08
61E8- 85 19 A0 00 B1 06 C5 19
61F0- F0 32 C8 B1 06 85 18 A2
61F8- 02 C6 18 E8 A5 18 C9 60
6200- F0 04 C9 40 D0 F3 18 A0
6208- 00 8A 65 06 85 06 A5 07
6210- 69 00 85 07 CD 01 40 D0
6218- D3 A5 06 CD 00 40 D0 CC
6220- A9 FF 85 08 60 A0 01 B1
6228- 06 85 18 A2 00 C8 B1 06
6230- 9D 00 03 E8 C8 C6 18 A5
6238- 18 C9 60 F0 04 C9 40 D0
6240- ED 8A C9 10 F0 09 A9 00
6248- 9D 00 03 E8 4C 41 62 60
6250- A2 00 A9 0B 85 24 A9 80
6258- 85 18 BD 00 03 85 19 A0
6260- 00 A9 10 85 25 A5 25 20
6268- C1 FB A5 28 18 65 24 85

```

```

6270- 28 A5 19 25 18 F0 05 A9
6278- AA 4C 7E 62 A9 A0 91 28
6280- C6 25 46 18 A5 18 C9 00
6288- D0 DB E6 24 E8 8A C9 10
6290- D0 C4 60 A0 01 B1 06 85
6298- 18 A2 02 C6 18 E8 A5 18
62A0- C9 60 F0 04 C9 40 D0 F3
62A8- 18 8A 65 06 85 08 A5 07
62B0- 69 00 85 09 A0 00 A5 08
62B8- CD 00 40 D0 0A A5 09 CD
62C0- 01 40 D0 03 4C F1 62 B1
62C8- 08 91 06 18 A5 06 69 01
62D0- 85 06 A5 07 69 00 85 07
62D8- 18 A5 08 69 01 85 08 A5
62E0- 09 69 00 85 09 CD 01 40
62E8- D0 DD A5 08 CD 00 40 D0
62F0- D6 A5 06 8D 00 40 A5 07
62F8- 8D 01 40 60 A2 00 BD 00
6300- 03 4A 9D 00 03 E8 E0 10
6308- D0 F4 60 A2 00 BD 00 03
6310- 0A 9D 00 03 E8 E0 10 D0
6318- F4 60 A2 0E BD 00 03 E8
6320- 9D 00 03 CA CA E0 00 D0
6328- F3 BD 00 03 E8 9D 00 03
6330- A9 00 8D 00 03 60 A2 01
6338- BD 00 03 CA 9D 00 03 E8
6340- E8 E0 0F D0 F3 BD 00 03
6348- CA 9D 00 03 A9 00 8D 0F
6350- 03 60

```

BYTE.FONT

```

4000-00 28 28 28 28 0D 0B 09 00 24 28 38 2C 2A 01 06
4010-00 28 28 13 2C 2A 01 06 00 28 28 17 29 2B 01 07
4020-00 10 10 17 39 03 01 07 00 70 28 2F 11 02 01 07
4030-00 28 18 28 05 07 05 02 00 08 2E 14 14 14 08 00

```

```

4040-00 18 20 12 00 02 03 02 00 10 10 15 2D 02 05 05
4050-00 08 08 17 09 29 01 01 00 10 10 10 2A 05 05 05
4060-00 08 18 09 29 03 01 02 00 28 18 28 19 01 01 06
4070-00 10 28 23 14 02 01 06 00 10 10 12 14 02 01 06
4080-00 28 08 08 08 05 05 02 00 10 10 10 12 05 05 02
4090-00 28 08 10 22 10 05 02 00 18 20 10 22 1D 05 02
40A0-00 20 28 28 22 25 05 03 00 28 18 28 09 08 08 09
40B0-00 10 10 28 2C 02 01 04 00 18 28 1F 28 18 01 07
40C0-00 24 24 22 25 01 01 06 00 28 28 2F 29 03 01 02
40D0-00 18 28 18 2C 1A 01 06 00 20 08 0F 21 03 01 02
40E0-00 18 20 11 09 22 01 02 00 18 20 16 00 22 01 06
40F0-00 18 20 15 00 22 05 02 00 18 20 12 00 25 05 05
4100-00 00 00 00 00 00 00 00 00 1C 00 1C 1C 1C 1C
4110-00 00 00 00 00 12 24 26 28 00 22 7F 22 22 22 2F 22
4120-00 08 28 20 2E 02 2E 08 00 21 52 24 08 12 25 42
4130-00 2E 12 0A 04 02 12 0C 00 00 00 0C 10 1C 1C
4140-00 18 0C 06 06 06 0C 18 0C 18 30 20 30 18 0C
4150-00 08 2A 1C 08 1C 2A 08 00 18 18 28 08 08 00
4160-00 10 1C 1C 00 00 00 00 00 00 00 22 00 00 00
4170-00 1C 1C 1C 00 00 00 00 01 02 04 08 10 20 40
4180-00 3E 26 26 22 22 22 2E 08 2E 08 08 08 08 0E
4190-00 28 06 06 2E 20 20 2E 00 2E 20 20 28 20 20 2E
41A0-00 20 30 7E 22 22 22 22 00 2E 22 20 2E 02 02 2E
41B0-00 2E 26 26 2E 02 22 2E 00 0C 0C 0C 0E 10 22 2E
41C0-00 2E 26 26 2E 22 22 2E 00 2E 22 20 2E 22 22 2E
41D0-00 00 1C 1C 00 1C 1C 00 0C 10 1C 1C 00 1C 1C
41E0-00 30 18 0C 06 0C 18 20 00 00 00 38 00 2E 00 00
41F0-00 06 0C 18 20 18 0C 06 04 00 08 78 40 42 2E
4200-00 3C 02 1A 3A 2A 22 1C 00 22 22 22 21 3F 11 1F
4210-00 2F 22 22 21 2F 11 1F 00 2F 22 02 02 01 21 2F
4220-00 1F 22 22 21 21 1F 00 2F 02 02 01 0F 01 2F
4230-00 02 02 01 1F 01 2F 00 2F 22 22 29 01 21 2F
4240-00 22 22 21 2F 21 21 00 0C 0C 0C 04 04 04
4250-00 1E 31 20 20 20 20 20 00 22 22 12 0F 09 11 21
4260-00 2F 02 02 01 01 01 00 48 48 48 49 49 49 4F
4270-00 22 28 28 29 29 29 22 00 2F 22 21 21 21 2F
4280-00 02 02 02 2F 21 21 2F 00 2F 12 28 29 21 21 2F
4290-00 22 22 12 2F 21 21 2F 00 2F 20 20 2F 01 01 2F
42A0-00 0C 0C 0C 04 04 04 2F 00 2F 22 22 21 21 21
42B0-00 08 14 22 22 22 21 21 00 7F 48 48 49 49 41
42C0-00 21 21 12 0C 0C 12 21 00 0C 0C 04 2F 21 21 21
42D0-00 2F 02 02 04 08 10 2F 00 3E 06 06 06 04 04 2E
42E0-00 40 20 10 09 04 02 01 00 2E 20 30 20 20 2E
42F0-00 00 00 62 26 1C 08 00 2E 00 00 00 00 00 00
4300-00 00 00 00 10 18 0C 06 00 2E 22 2E 20 2E 00
4310-00 2E 22 22 2E 02 02 00 2E 02 02 02 0E 00 00
4320-00 2E 22 22 2E 20 20 00 2E 02 2E 22 2E 00 00
4330-00 04 04 04 1E 04 2C 2E 20 2E 22 22 2E 00 00
4340-00 22 22 22 2E 02 02 00 08 08 08 08 08 04
4350-1E 12 10 10 10 10 10 00 22 12 0E 12 22 02 02
4360-00 08 08 08 08 08 08 08 08 49 49 49 49 7F 00
4370-00 22 22 22 22 2E 00 00 00 2E 22 22 22 2E 00
4380-02 02 2E 22 22 2E 00 00 20 2E 22 22 2E 00 00
4390-00 02 02 02 22 2E 00 00 2E 20 2E 02 2E 00 00
43A0-00 2C 04 04 04 2E 04 00 2E 22 22 22 22 00 00
43B0-00 08 14 22 22 22 00 00 7F 49 49 49 49 00 00
43C0-00 22 14 08 14 22 00 00 2E 20 2E 22 22 00 00
43D0-00 2E 04 08 10 2E 00 00 18 0C 0C 06 0C 0C 18
43E0-08 08 08 08 08 08 08 00 18 30 20 60 20 20 18
43F0-00 02 06 0E 1E 0E 02 2F 2F 7F 2F 7F 7F 7F
4400-8F 80 FF FF 00 00 00 00

```

```

ABCDEFGHIJKLMNOPQRSTUVWXYZ 01
abcdefghijklmnopqrstuvwxyz !@

```

Programme CREATE

```

100 HIMEM: 16383:D$ = CHR$(4):CR$ = CHR$(13):
PRINT : PRINT CHR$(27): PRINT CHR$(17): PRINT
: PRINT CR$;D$;"PR#0":FLAG = 0
105 P01 = 24576:P10 = 24609:P20 = 24612:P30 =
24659:P40 = 24719
110 P50 = 24790:P60 = 25054:S65 = 25125:P70 =
25168:P80 = 25235
115 S10 = 25340:S20 = 25355:S30 = 25370:S40 = 25398
120 PRINT CR$: PRINT D$;"BLOAD
SAISIE.OBJ,A$6000"
125 ONERR GOTO 220
130 REM ----- MENU -----
135 HOME : VTAB 6: HTAB 1: PRINT "<0> FIN DE
PROGRAMME
140 VTAB 8: HTAB 1: PRINT "<1> CREATION D'UN JEU
DE CARACTERE": VTAB 9: HTAB 20: PRINT
"NOUVEAU FICHIER"
145 VTAB 11: HTAB 1: PRINT "<2> AJOUT DE
NOUVEAUX CARACTERES A UN": VTAB 12: HTAB
20: PRINT "FICHIER ANCIEN"
150 VTAB 14: HTAB 1: PRINT "<3> AFFICHAGE/MODIF

```

```

DE CARACTERES SUR UN": VTAB 15: HTAB 20:
PRINT "FICHIER ANCIEN"
155 VTAB 17: HTAB 1: PRINT "<4> RECUPERATION
D'UN JEU DE CARACTERE": VTAB 18: HTAB 20:
PRINT "CHAT MAUVE"
160 VTAB 20: HTAB 1: PRINT "<5> CHARGEMENT D'UN
FICHIER SUR PRINTER": CALL - 958
165 VTAB 23: HTAB 10: PRINT "VOTRE CHOIX ": GET
R$
170 IND = 1: GOSUB 870: IF IND = 5 THEN NFRM = 2
175 ON IND GOTO 165,180,285,185,295,225,185
180 END
185 REM ----- FICHIER ANCIEN -----
190 VTAB 23: HTAB 10: INPUT "NOM DU FICHIER ?
":NOM$
195 IF NOM$ = "?" THEN PRINT CHR$(4)"CATALOG":
PRINT : PRINT : VTAB 23: HTAB 1: CALL - 958:
GOTO 190
200 PRINT CHR$(4);"BLOAD"NOM$.CHAR,A$4000"
205 POKE 8, PEEK (16384): POKE 9, PEEK (16385)
210 IF ASC (R$) = 50 THEN NFRM = 1: GOTO 295
215 GOTO 750

```

```

220 PRINT "FICHIER NON TROUVE...Appuyez sur
RETURN "; GET R$: GOTO 130
225 REM ----- FICHIER CHAT MAUVE -----
230 VTAB 23: HTAB 10: INPUT "NOM DU FICHIER ?
";NOM$: IF NOM$ = "?" THEN PRINT CHR$(
4)"CATALOG": PRINT : PRINT : VTAB 23: HTAB 1:
CALL - 958: GOTO 230
235 PRINT CHR$(4);"BLOAD"NOM$.FONT,A$4000"
240 CALL P50
245 VTAB 20: HTAB 1: CALL - 958
250 VTAB 22: HTAB 1: PRINT "1 AFFICHAGE / 0
SAUEGARDE": VTAB 23: HTAB 1: PRINT "
MODIFICATION ENVOI PRINTER "; GET R$
255 IND = 1: GOSUB 890
260 ON IND GOTO 250,275,265
265 NFRM = 3
270 GOTO 295
275 POKE 8, PEEK (16384): POKE 9, PEEK (16385)
280 GOTO 750
285 REM ----- CONSTRUCTION DU CADRE -----
290 NFRM = 0
295 HOME : IF NFRM < 2 THEN VTAB 1: HTAB 1: PRINT
"MATRICE IMPRIMANTE - HAUTEUR : 9 ": VTAB 2:
HTAB 1: PRINT "LARGEUR DE 1 A 16 ": PRINT
"MAJUSCULES : A...P": VTAB 3: HTAB 19: PRINT
"minuscules : a...p"
300 FOR I = 1 TO 16: VTAB 8: HTAB (11 + I): PRINT
CHR$(64 + I): VTAB 9: HTAB (11 + I): PRINT "-":
VTAB 18: HTAB (11 + I): PRINT "-": VTAB 19: HTAB
(11 + I): PRINT CHR$(96 + I): NEXT I
305 FOR I = 1 TO 8: HTAB 10: VTAB 9 + I: PRINT I:"I":
VTAB 9 + I: HTAB 28: PRINT "I": PRINT I + 1: NEXT I
310 IF NFRM = 2 THEN 400
315 IF NFRM = 3 THEN 425
320 REM ----- DEPLACEMENT DU CURSEUR -----
325 VTAB 20: HTAB 1: CALL - 958: VTAB 21: HTAB 1:
PRINT "FLECHES : DEPLACEMENT DU CURSEUR":
VTAB 22: HTAB 1: PRINT "ESPACE : CHANGEMENT
0 : FIN "
330 XD = 12: YD = 10
335 HTAB XD: VTAB YD: GET R$
340 IF ASC (R$) = 8 THEN XD = XD - 1: GOTO 365
345 IF ASC (R$) = 10 THEN YD = YD + 1: GOTO 365
350 IF ASC (R$) = 11 THEN YD = YD - 1: GOTO 365
355 IF ASC (R$) = 21 THEN XD = XD + 1: GOTO 365
360 IF R$ = "?" THEN CALL P40: GOTO 680
365 IF XD < 12 THEN XD = 12
370 IF XD > 27 THEN XD = 27
375 IF YD < 10 THEN YD = 10
380 IF YD > 17 THEN YD = 17
385 IF ASC (R$) = 48 OR ASC (R$) = 64 THEN CALL
P40: GOTO 580
390 IF ASC (R$) = 32 THEN CALL P01
395 GOTO 335
400 REM ---- AFFICHAGE CARACTERE EXISTANT ----
405 VTAB 4: HTAB 1: INPUT "NOM DU FICHIER ? ";NOM$
410 IF NOM$ = "?" THEN CALL - 958: PRINT CHR$(
4)"CATALOG": PRINT : PRINT : VTAB 23: HTAB 1:
CALL - 958: PRINT "TAPEZ SUR UNE TOUCHE":
GET R$: GOTO 295
415 PRINT CHR$(
4)"BLOAD"NOM$.CHAR,A$4000":NFRM = 3:NR =
0: GOTO 245
420 VTAB 4: HTAB 1: PRINT "NOM DU FICHIER ? ";NOM$

```

```

425 VTAB 6: HTAB 1: PRINT "CARACTERE A
RECHERCHER ? "; GET O$: PRINT O$
430 IF ASC (O$) < > 13 THEN 465
435 IF NR = 0 THEN NR = 1: Q$ = CHR$ ( PEEK (16394)):
GOTO 460
440 DEP = PEEK ( PEEK (7) * 256 + PEEK (6) + 1) - 64:
IF DEP > 32 THEN DEP = DEP - 32
445 DEP = DEP + PEEK (7) * 256 + PEEK (6) + 2
450 IF DEP > = PEEK (16384) + PEEK (16385) * 256
THEN NR = 0: GOTO 435
455 Q$ = CHR$ ( PEEK (DEP))
460 VTAB 6: HTAB 1: CALL - 868: PRINT "CARACTERE
SUR ECRAN : ";Q$
465 IF PEEK (16384) = 10 AND PEEK (16385) = 64 THEN
VTAB 21: HTAB 1: PRINT "LE FICHIER NE
CONTIENT AUCUN ELEMENT.": VTAB 22: HTAB 1:
PRINT "..RETOUR AU MENU..Appuyez sur
RETURN.": GET R$: GOTO 130
470 NR = 1: POKE 8, ASC (O$): CALL P60
475 IF PEEK (8) < > 255 THEN 500
480 VTAB 22: HTAB 1: PRINT "NON TROUVE":NR = 0
485 VTAB 23: HTAB 1: PRINT "UNE AUTRE RECHERCHE
O/N ? "; GET R$: PRINT R$: IF R$ = "O" OR R$ = "o"
THEN HOME : GOTO 295
490 IF R$ < > "N" AND R$ < > "n" THEN VTAB 20: HTAB
1: CALL - 958: GOTO 485
495 GOTO 245
500 CALL S65: CALL P70: REM AFFICHAGE
505 VTAB 22: HTAB 1: PRINT "0: FIN
1: SUPPRESSION": VTAB 23: HTAB 1: PRINT "2:
MODIFICATION 3: AUTRE CARACTERE "; GET R$
510 IF R$ = "?" THEN 680
515 IND = 1: GOSUB 880
518 IF ASC (R$) = 13 THEN 435
520 ON IND GOTO 505,245,525,320,295
525 VTAB 22: HTAB 1: CALL - 958: CALL P80: FLAG =
1: NR = 0: GOTO 485: REM SUPPRESSION
530 POKE 8, ASC (R$): CALL P60: FLAG = 1: IF PEEK (8)
= 255 THEN 540
535 CALL P80: REM SUPPRESSION
540 R = ASC (R$): POKE 6, R: R = ASC (L$): POKE 7, R
545 R = R - 64: IF R > 32 THEN R = R - 32
550 POKE 24, R
555 POKE 8, PEEK (16384): POKE 9, PEEK (16385)
560 CALL P20: REM
565 GOTO 485
570 IF FLAG = 0 THEN 245
575 GOTO 750
580 REM -- CHARGT CODE A $4000 ET SUIVANTS-----
585 VTAB 20: HTAB 1: CALL - 958: VTAB 21: HTAB 1
590 IF NFRM < > 3 THEN 610
595 PRINT "MEME CODE ASCII <";Q$;"> O/N ? "; GET
R$: PRINT R$: IF R$ = "O" OR R$ = "o" THEN R$ =
Q$: GOTO 625
600 IF R$ < > "N" AND R$ < > "n" THEN 585
605 VTAB 20: HTAB 1: CALL - 958: VTAB 21: HTAB 1
610 PRINT "CODE CARACTERE DU DESSIN ? "; GET
R$: PRINT R$: IF ASC (R$) = 13 AND NFRM = 3
THEN 505
615 IF ASC (R$) = 13 THEN 320
620 IF ASC (R$) < 26 OR ASC (R$) > 127 THEN 585
625 VTAB 22: HTAB 1: PRINT "LARGEUR DU CARATERE
? "; GET L$: PRINT L$: IF ASC (L$) = 13 AND NFRM
= 3 THEN 505

```

```

630 IF ASC (L$) = 13 THEN 320
635 IF ( ASC (L$) > 64 AND ASC (L$) < 81) OR ( ASC (L$)
    > 96 AND ASC (L$) < 113) THEN 645
640 GOTO 625
645 IF NFRM = 3 THEN 530
650 R = ASC (R$); POKE 6,R;R = ASC (L$); POKE 7,R
655 R = R - 64; IF R > 32 THEN R = R - 32
660 POKE 24,R
665 IF NFRM = 0 THEN NFRM = 1; CALL P10; GOTO 725
670 POKE 8, PEEK (16384); POKE 9, PEEK (16385);
    CALL P20
675 GOTO 725
680 REM ----- DECALAGES -----
685 VTAB 22: HTAB 1: PRINT "DECALAGES 0: HAUT
    1: BAS "; VTAB 23: HTAB 1: PRINT "<CR> RETOUR
    2: DROITE 3: GAUCHE "; GET R$
690 IF ASC (R$) = 13 THEN 505
695 IND = 1; GOSUB 880
700 ON IND GOTO 685,705,710,715,
    720
705 CALL S10; CALL P70; GOTO 320
710 CALL S20; CALL P70; GOTO 320
715 CALL S30; CALL P70; GOTO 320
720 CALL S40; CALL P70; GOTO 320
725 REM ----- CONTINUATION -----
730 VTAB 23: HTAB 1: CALL - 958; IF NFRM = 3 THEN
    465
735 PRINT "NOUVELLE FORME O/N ? "; GET R$; PRINT
    R$; IF R$ = "O" OR R$ = "o" THEN HOME : GOTO 295
740 IF R$ < > "N" AND R$ < > "n" THEN 730
745 GOTO 245
750 REM ----- SAVE FICHER -----
755 VTAB 20: HTAB 1: CALL - 958
760 VTAB 22: HTAB 1: PRINT "SAUVEGARDE DU
    FICHER O/N ? "; GET R$; PRINT R$; IF R$ = "N" OR
    R$ = "n" THEN 780
765 IF R$ < > "O" AND R$ < > "o" THEN 760
770 IF NOM$ = "" THEN 773
771 VTAB 22: HTAB 1: PRINT "SAUVEGARDE SOUS LE
    NOM DE <;NOM$;> O/N "; GET R$; PRINT R$; IF
    R$ = "O" OR R$ = "o" THEN 775
772 IF R$ < > "N" AND R$ < > "n" THEN 760
773 VTAB 23: HTAB 1: INPUT "NOM DU FICHER ?
    ";NOM$
775 A = 4 * 16 ^ 3;L = PEEK (16384) + PEEK (16385) *
    256 - A; PRINT CHR$
    (4)"BSAVE" "NOM$".CHAR,A,"A",L,L
780 REM ----- ENVOI VERS PRINTER -----
785 VTAB 22: HTAB 1: CALL - 958
790 PRINT "ENVOI VERS PRINTER O/N ? "; GET R$;
    PRINT R$; IF R$ = "N" OR R$ = "n" THEN 885
795 IF R$ < > "O" AND R$ < > "o" THEN 780
800 POKE 8, PEEK (16384); POKE 9, PEEK (16385)
805 PRINT CR$;D$;"PR#1"; PRINT CR$; CALL P30;
    PRINT CHR$ (4);"PR#0"
810 VTAB 23: HTAB 1: PRINT "CONTROLE PRINTER
    O/N ? "; GET R$; PRINT R$; IF R$ = "N" OR R$ = "n"
    THEN 860
815 IF R$ < > "O" AND R$ < > "o" THEN 810
820 PRINT CR$;D$;"PR#1"; PRINT CHR$ (27); CHR$ (39)
825 PRINT "ABCDEFGHJKLMNOPQRSTUVWXYZ
    0123456789 -=";"$;"$";/
830 PRINT CHR$ (27); CHR$ (36)
835 PRINT "ABCDEFGHJKLMNOPQRSTUVWXYZ
    0123456789 -=";"$;"$";/
840 PRINT CHR$ (27); CHR$ (39)
845 PRINT "abcdefghijklmnopqrstuvwxy
    zlà#$$%^&*()_+éè:~ù<>?"
850 PRINT CHR$ (27); CHR$ (36)
855 PRINT "abcdefghijklmnopqrstuvwxy
    zlà#$$%^&*()_+éè:~ù<>?"
860 PRINT CHR$ (4);"PR#0"
865 GOTO 130
870 IF ASC (R$) = 53 OR ASC (R$) = 40 THEN IND = 7
875 IF ASC (R$) = 52 OR ASC (R$) = 39 THEN IND = 6
880 IF ASC (R$) = 51 OR ASC (R$) = 34 THEN IND = 5
885 IF ASC (R$) = 50 OR ASC (R$) = 123 THEN IND = 4
890 IF ASC (R$) = 49 OR ASC (R$) = 38 THEN IND = 3
895 IF ASC (R$) = 48 OR ASC (R$) = 64 THEN IND = 2
900 RETURN

```

Hard copy de la page HGR sur ImageWriter

Yvan Koenig

Il existe un moyen simple d'obtenir une copie papier des pages graphiques de l'Apple 280 x 192 points à partir du programme de Marc Debui-gne du Pom's 18. Il s'agit d'un patch utile à ceux qui ne disposent pas d'une carte 80 colonnes étendue.

Si vous possédez l'assembleur Big Mac, il suffit dans le source de supprimer les lignes 97, 101 à 104, 143 à 148. Il faut en outre remplacer la ligne 132 par LDA #\$B0, la 134 par LDA #\$B5, et la 136 par LDA #\$B6.

Pour modifier directement le code objet il faut pratiquer comme suit :

```

BLOAD HARD
CALL -151
405D:EA EA EA
4069:EA EA EA EA EA EA
4070:EA EA EA EA EA
40BB:B0
40C0:B5
40C5:B6
40D6:EA EA
40D8:EA EA EA EA EA EA EA EA
40E0:EA EA EA EA EA EA EA
BSAVE HARD,A$4000,L$1A0

```

Pour respecter la géométrie de l'image à reproduire, il conviendra de n'imprimer qu'en taille normale et de

mettre auparavant l'imprimante en mode dilaté par :

```

PRINT D$"PR#1";
PRINT CHR$(14)

```

Un Bug ?

Essayez :

```

10 INPUT A : PRINT A
20 GOTO 20

```

avec les valeurs 5,032 et 6,032. Qui peut expliquer ce phénomène ?

Mr Baron - 31530 Lévigac.

Retrouvez vos programmes perdus

Guy d'Herbemont

Nul n'est à l'abri d'une erreur de manipulation : vérité première qui peut parfois porter gravement atteinte à la productivité du programmeur !

Je pense là plus particulièrement à un type d'erreur que vous avez vraisemblablement tous rencontré, hélas, au moins une fois. Au cours de la mise au point d'un programme Applesoft, vous avez ainsi sauvé sur disquette une première version de votre travail, comportant quelques lignes de Basic, sous le nom de PGM, par exemple. Peu de temps après, vous attaquez la suite de la tâche avec énergie, améliorez les premières instructions, rentrez en machine l'essentiel du programme et, consciencieusement, songez à sauvegarder le fruit de vos efforts. Et c'est alors que, victime de la fatigue, vous tapez "LOAD PGM" en lieu et place de "SAVE PGM", réduisant ainsi à néant plusieurs heures de labeur, pour ne plus retrouver en mémoire que les quelques lignes de départ...

Erreur funeste, il est vrai, mais pas irréparable, car la suite de votre programme se trouve toujours en mémoire, mais au delà du pointeur de fin de programme, ce qui vous empêche de la retrouver, tant pour un LIST que pour un SAVE.

La petite routine TROUVE que je vous propose ici, permet justement de reconstituer une fin de programme ainsi perdue, en modifiant les pointeurs de début et de fin de programme de telle façon que la partie de mémoire délimitée par ces pointeurs présente à nouveau une structure correcte tant pour le DOS que pour l'Applesoft.

Les principes utilisés sont les suivants :

- En partant de l'adresse donnée par le pointeur de fin de programme initial, chercher le premier 0 qui puisse être correctement

considéré comme marqueur de fin de ligne.

- Le premier octet suivant ce 0 correspond au début de la première ligne Applesoft récupérable, et son adresse sera donnée comme nouveau pointeur de début de programme.
- On parcourt ensuite les lignes successives, jusqu'à trouver si possible les deux zéros marquant la fin de programme, et ce en mettant à jour le pointeur de fin de programme en conséquence.

Si cette routine est exécutée victorieusement, vous pouvez ensuite sauver la partie de programme ainsi reconstituée. Par le biais de sa transformation ultérieure en fichier EXEC, par exemple, il est permis de la fusionner avec le début de programme déjà sauvegardé. Il ne reste alors qu'à "boucher" un petit trou entre les deux parties et à reprendre les modifications éventuelles sur les premières lignes, avant de sauver (sans erreur si possible...) le programme complet ainsi retrouvé.

Il est possible toutefois que la routine débouche sur un échec, lorsque les octets situés au delà du pointeur de programme initial (après le LOAD intempestif) ne permettent pas de retrouver une structure Applesoft cohérente. Ceci sera souvent le cas si d'autres opérations, telles que RUN, sont effectuées avant que l'erreur ne soit constatée et que la routine soit utilisée. Notons par ailleurs que TROUVE ne permettra jamais, bien sûr, de récupérer un programme écrasé par le chargement d'un autre plus long que lui, car la perte est cette fois effective et ne se situe plus du tout au niveau du pointeur de fin de programme.

Mode d'emploi

Lorsque vous constatez l'erreur de manipulation, chargez de suite la

routine par BLOAD TROUVE.OBJ et lancez-en l'exécution par CALL 768. S'il n'y a pas de message "ECHEC", vous pouvez lister la partie de programme récupérée et la sauver sur disquette.

Par un CALL 863 (utilisation de la sous-routine ECHEC), remettez ensuite les pointeurs de début et de fin de programme à leurs valeurs standard.

Rechargez alors la partie de programme récupérée et transformez-la en fichier EXEC (voir manuel du DOS 3.3).

Faites enfin un LOAD du début de programme, puis un EXEC de la seconde partie, et terminez les opérations normalement sur le programme complet reconstitué en mémoire.

Récapitulation TROUVE.OBJ

```
0300- A5 AF 85 67 A5 B0 85 68
0308- A5 B0 C9 96 B0 51 A0 00
0310- B1 67 F0 08 E6 67 D0 F8
0318- E6 68 D0 F4 E6 67 D0 02
0320- E6 68 A5 67 85 AF A5 68
0328- 85 B0 A0 01 B1 AF F0 19
0330- C5 B0 90 D4 E6 B0 C5 B0
0338- F0 02 B0 CC 48 88 C6 B0
0340- B1 AF 85 AF 68 85 B0 D0
0348- E1 88 B1 AF D0 BA A5 AF
0350- 18 69 03 85 AF A5 B0 69
0358- 00 85 B0 20 8B FD 60 A9
0360- 01 85 67 A9 08 85 68 85
0368- B0 A0 03 84 AF A9 00 88
0370- 91 67 D0 FB 20 8B FD A2
0378- 0A BD 86 03 20 ED FD CA
0380- D0 F7 20 8B FD 60 20 03
0388- 20 05 20 08 20 03 20 05
0390- 20 16
```

Programme TROUVE.SCE (Assembleur LISA 1.5)

```
1      ORG $300
2 PRGDEB EPZ $67
3 PRGEND EPZ $AF
4 COUT  EQU $FDED
5 CROUT1 EQU $FD8B
```

```
6 LIM EQU $96 ;page fin recherche
7 LDA PRGEND
8 STA PRGDEB
9 LDA PRGEND+1
10 STA PRGDEB+1
11 DEBUT LDA PRGEND+1 ;cherch 1er octet nul
12 CMP #LIM ;pas au delà de LIM
13 BCS ECHEC
14 LDY #0
```

```

15 D1      LDA (PRGDEB),Y
16          BEQ D2          ;octet trouvé
17          INC PRGDEB
18          BNE D1
19          INC PRGDEB+1
20          BNE D1          ;sinon octet suivant
21 D2      INC PRGDEB
22          BNE D3
23          INC PRGDEB+1
24 D3      LDA PRGDEB
25          STA PRGEND
26          LDA PRGDEB+1
27          STA PRGEND+1
28 SUITE   LDY #1          ;chercher la fin
                        d'adresse en adresse
29          LDA (PRGEND),Y;POIDS FORT
30          BEQ TEST1      ;fin ?
31          CMP PRGEND+1
32          BCC DEBUT      ;adresse trop basse
33          INC PRGEND+1
34          CMP PRGEND+1
35          BEQ S1
36          BCS DEBUT      ;adresse trop élevée
37 S1      PHA
38          DEY
39          DEC PRGEND+1
40          LDA (PRGEND),Y;Poids faible de
41          STA PRGEND      ;l'adresse suivante
42          PLA
43          STA PRGEND+1    ;et son poids fort
44          BNE SUITE      ;branche toujours
45 TEST1   DEY
46          LDA (PRGEND),Y
47          BNE DEBUT
48 FIN     LDA PRGEND
49          CLC
50          ADC #3
51          STA PRGEND
52          LDA PRGEND+1
53          ADC #0
54          STA PRGEND+1
55          JSR CROUT1
56          RTS
57 ECHEC   LDA #1
58          STA PRGDEB
59          LDA #8
60          STA PRGDEB+1
61          STA PRGEND+1
62          LDY #3
63          STY PRGEND
64          LDA #0
65 E1      DEY
66          STA (PRGDEB),Y
67          BNE E1
68          JSR CROUT1
69          LDX #$0A
70 E2      LDA TEXT2,X
71          JSR COUT
72          DEX
73          BNE E2
74          JSR CROUT1
75          RTS
76 TEXT2   INV " C E H C E "
77          DCM "INT"
78          END

```

Erratum

Lors du montage de "Gestion de fenêtres texte", programme de G. Zwingelstein paru dans le numéro 18, un coup de ciseaux malheureux a supprimé les 4/5 du source assembleur et de la récapitulation. Vous trouverez ci-dessous le code hexadécimal complet. Le source, trop important pour une publication intégrale n'était pas non plus sur la disquette d'accompagnement. Nous proposons à ceux d'entre vous qui possèdent BigMac de nous envoyer une disquette vierge et une enveloppe self-adressée affranchie à 3,20 F. Nous renverrons l'intégralité du programme.

A noter que le code objet de la disquette d'accompagnement suffit à une utilisation complète de ce programme.

```

89EE- A9 4C
89F0- 8D F5 03 A9 02 8D F6 03
89F8- A9 8A 8D F7 03 4C 59 FA
8A00- 00 8A 48 20 B1 00 68 A2
8A08- 0F DD 20 8A F0 06 CA D0
8A10- F8 4C C9 DE CA 8A 0A AA
8A18- BD 31 8A 48 BD 30 8A 48
8A20- 60 A2 96 AA B6 B7 A8 A7
8A28- A3 49 43 4F 46 45 44 50
8A30- AC 8A C5 8A 2F 91 84 92
8A38- 44 92 04 93 41 93 55 8B
8A40- 58 8A 24 8F 4A 8F 69 8F
8A48- 0B 90 8A 8F A4 8F 00 00
8A50- 00 00 00 00 12 8A 00 00
8A58- 00 38 AD 00 8A E9 08 85
8A60- FA AD 01 8A E9 00 85 FB
8A68- A9 00 8D 4E 8A 8D 4F 8A
8A70- 8D 52 8A 20 B9 95 A9 28
8A78- 2C 58 8A 10 01 0A 8D 53
8A80- 8A A9 00 8D 54 8A A9 18
8A88- 8D 55 8A 20 AB 8D 20 C6
8A90- 8D A6 FA A5 FB 86 73 85
8A98- 74 20 6C D6 A6 76 E8 F0
8AA0- 09 A6 33 E0 DD F0 03 4C
8AA8- D2 D7 4C 3C D4 20 F8 E6
8AB0- CA E0 18 B0 7F 8A 18 65
8AB8- 22 C5 23 80 77 85 25 8D
8AC0- FB 05 20 A7 95 60 20 F8
8AC8- E6 CA E4 21 B0 66 8A 85
8AD0- 24 8D 7B 05 8D 7B 04 60
8AD8- 20 E7 8D 20 21 8D B0 57
8AE0- 20 24 8E 20 60 8D A5 73
8AE8- 85 FA A5 74 85 FB 20 C8
8AF0- 8C 90 3E 60 20 E7 8D 20
8AF8- 21 8D 90 40 48 AD 56 8A
8B00- 20 07 8D 90 3A 68 60 20
8B08- E7 8D 20 21 8D 90 2D AD
8B10- 56 8A 20 07 8D B0 28 60
8B18- A9 00 20 07 8D F0 1D 60
8B20- 20 E7 8D 20 07 8D B0 0C
8B28- AD 56 8A 20 21 8D 90 0C
8B30- 60 4C 10 D4 4C 99 E1 A2
8B38- 78 4C 12 D4 4C 7C D9 4C
8B40- D5 E8 20 E7 8D 20 07 8D
8B48- 08 AD 56 8A 20 21 8D 90
8B50- EB 20 F3 8D 28 60 A9 00
8B58- 20 21 8D 18 A9 08 65 FA
8B60- 85 FA 90 02 E6 FB 20 67
8B68- DD 20 52 E7 38 A5 FA E5
8B70- 50 85 71 A5 FB E5 51 85
8B78- 72 20 C8 8C 90 B3 A5 FB
8B80- C5 70 D0 06 A5 FA C5 6F
8B88- F0 30 A5 6F 85 3C A5 70
8B90- 85 3D A4 FB A6 FA D0 01
8B98- 88 CA 86 3E 84 3F A5 5E
8BA0- 85 42 A5 5F 85 43 A0 00
8BA8- A5 6F C5 5E A5 70 E5 5F
8BB0- 90 05 20 2C FE B0 03 20
8BB8- 8B 8C A5 5E 85 6F A5 5F
8BC0- 85 70 A5 50 85 73 A5 51
8BC8- 85 74 A9 55 A2 00 85 5E
8BD0- 86 5F C5 52 F0 05 20 51
8BD8- 8C F0 F7 A9 07 85 8F A5
8BE0- 69 A6 6A 85 5E 86 5F E4
8BE8- 6C D0 04 C5 6B F0 05 20
8BF0- 47 8C F0 F3 85 94 86 95
8BF8- A9 03 85 8F A5 94 A6 95
8C00- E4 6E D0 05 C5 6D D0 01
8C08- 60 85 5E 86 5F A0 00 B1
8C10- 5E AA C8 B1 5E 08 C8 B1
8C18- 5E 65 94 85 94 C8 B1 5E
8C20- 65 95 85 95 28 10 D5 8A
8C28- 30 D2 C8 B1 5E A0 00 0A
8C30- 69 05 65 5E 85 5E 90 02
8C38- E6 5F E4 95 D0 04 C5 94
8C40- F0 BE 20 51 8C F0 F3 B1
8C48- 5E 30 30 C8 B1 5E 10 2B
8C50- C8 B1 5E F0 26 C8 38 B1
8C58- 5E E5 71 AA C8 B1 5E E5
8C60- 72 C5 70 90 16 D0 04 E4
8C68- 6F 90 10 C5 74 90 06 D0
8C70- 0A E4 73 B0 06 91 5E 88
8C78- 8A 91 5E A5 8F 18 65 5E
8C80- 85 5E 90 02 E6 5F A6 5F
8C88- A0 00 60 38 A5 3E 3E 3C
8C90- 85 40 A5 3F E5 3D 85 41
8C98- 18 A5 42 65 40 85 40 A5
8CA0- 43 65 41 85 41 B1 3E 91
8CA8- 40 20 AF 8C 90 F7 60 A6
8CB0- 40 D0 02 C6 41 C6 40 A5
8CB8- 3C C5 3E A5 3D E5 3F A6
8CC0- 3E D0 02 C6 3F C6 3E 60
8CC8- 20 84 E4 38 A5 73 E5 71
8CD0- 85 50 A5 74 E5 72 85 51
8CD8- 38 A5 FA E5 71 85 5E A5
8CE0- FB E5 72 85 5F AD 00 8A
8CE8- C5 5E AD 01 8A E5 5F 90
8CF0- 15 38 A5 6F E5 71 85 5E
8CF8- A5 70 E5 72 85 5F A5 5E
8D00- C5 6D A5 5F E5 6E 60 48
8D08- A9 00 20 21 8D F0 0C 68
8D10- CD 4F 8A F0 08 48 AD 4F
8D18- 8A D0 EF 68 18 AD 4E 8A
8D20- 60 48 A5 73 85 FA A5 74
8D28- 85 FB 20 B9 8D 68 CD 4E
8D30- 8A F0 1B 48 AD 4E 8A F0
8D38- 13 20 60 8D 18 A5 FA 65
8D40- 71 85 FA A5 FB 65 72 85
8D48- FB 4C 2A 8D 68 18 AD 4F
8D50- 8A 60 A9 08 18 65 FA 85
8D58- FC A5 FB 69 00 85 FD 60
8D60- A9 08 85 71 A9 00 F0 04
8D68- A9 00 85 71 85 72 38 AD
8D70- 55 8A ED 54 8A F0 10 88
8D78- AD 53 8A 18 65 71 85 71
8D80- 90 02 E6 72 88 D0 F1 60
8D88- A5 25 38 E5 22 8D 50 8A
8D90- A5 24 2C 58 8A 10 03 AD
8D98- 7B 05 8D 51 8A 60 AD 50
8DA0- 8A 20 B6 8A AD 51 8A 20
8DA8- CF 8A 60 A0 04 B9 4E 8A
8DB0- 99 1C 00 C8 C0 08 90 F5
8DB8- 60 A0 00 B1 FA 99 4E 8A
8DC0- C8 C0 08 90 F6 60 A0 00
8DC8- B9 4E 8A 91 FA C8 C0 08
8DD0- 90 F6 60 A0 04 B9 4E 8A
8DD8- 48 B1 FA 99 4E 8A 68 91
8DE0- FA C8 C0 08 90 EF 60 20
8DE8- F8 E6 8A F0 34 30 32 8E
8DF0- 56 8A 60 20 BE DE 20 F8
8DF8- E6 CA 8A 2C 58 8A 10 02
8E00- 29 FE 8D 52 8A 20 BE DE
8E08- 20 F8 E6 CA 8A 38 ED 54
8E10- 8A 18 6D 55 8A 8D 55 8A

```

8E18- 8E 54 8A 20 78 8E 90 01
8E20- 60 4C 99 E1 A0 04 98 48
8E28- 20 BE DE 20 F8 E6 68 A8
8E30- 8A 99 4E 8A C8 C0 08 90
8E38- ED CE 54 8A CE 52 8A 2C
8E40- 58 8A 10 13 EE 53 8A AD
8E48- 53 8A 29 FE 8D 52 8A AD
8E50- 52 8A 29 FE 8D 52 8A AD
8E58- 53 8A ED 52 8A 8D 53 8A
8E60- 20 78 8E 90 BC 20 B7 00
8E68- F0 0A 20 BE DE 20 F8 E6
8E70- 8E 57 8A 60 A2 A0 D0 F8
8E78- AD 54 8A 30 33 CD 55 8A
8E80- B0 2E A9 18 CD 55 8A 90
8E88- 27 AD 52 8A 30 22 2C 58
8E90- 8A 10 0D 6A B0 1A AD 53
8E98- 8A 6A B0 14 A9 50 D0 02
8EA0- A9 28 38 ED 52 8A F0 08
8EA8- 90 06 ED 53 8A 90 01 60
8EB0- 18 60 20 52 8D AE 54 8A
8EB8- E8 AD 57 8A 20 D6 8E AD
8EC0- 53 8A 18 65 FC 85 FC 90
8EC8- 02 E6 FD A9 A0 E8 EC 55
8ED0- 8A 90 E9 F0 E4 60 48 A0
8ED8- 00 AD 57 8A 91 FC 68 4C
8EE0- E4 8E 91 FC C8 CC 53 8A
8EE8- D0 F8 88 AD 57 8A 91 FC
8EF0- 60 20 52 8D A5 22 48 20
8EF8- A7 95 A0 00 B1 FC AA B8
8F00- 20 53 95 91 FC 8A 2C 24
8F08- 8F 20 53 95 C8 C4 21 90
8F10- EB 18 A5 21 65 FC 85 FC
8F18- 90 02 E6 FD 18 68 69 01
8F20- C5 23 90 D2 60 20 D8 8A
8F28- 20 7E 8B A5 73 85 FA 8A
8F30- 74 85 FB AD 56 8A 8D 4E
8F38- 8A A9 00 8D 4F 8A 8D 50
8F40- 8A 8D 51 8A 20 B2 BE 20
8F48- C6 8D 60 20 07 8B 20 88
8F50- 8D AD 56 8A 8D 4F 8A 20
8F58- C6 8D AD 56 8A 20 21 8D
8F60- 20 AB 8D 20 F1 8E 20 9E
8F68- 8D 60 20 18 8B 20 88 8D
8F70- 20 F1 8E 20 C6 8D AD 4E
8F78- 8A 20 07 8D A9 00 8D 4F
8F80- 8A 20 AB 8D 20 9E 8D 20
8F88- C6 8D 60 20 20 8B 20 60
8F90- 8D 38 A9 00 E5 71 85 71
8F98- A9 00 E5 72 85 72 20 C8
8FA0- 8C 20 7E 8B 60 20 42 8B
8FA8- B0 04 20 D3 8D 60 A2 03
8FB0- BD 52 8A 48 CA 10 F9 AD
8FB8- 56 8A 8D 4F 8A 20 64 90
8FC0- A9 00 20 07 8D AD 56 8A
8FC8- CD 4E 8A 08 20 6D 8F 28
8FD0- D0 EE AD 56 8A 20 21 8D
8FD8- A2 00 68 9D 52 8A E8 E0
8FE0- 04 D0 F7 20 D3 8D A2 FF
8FE8- BD 00 02 F0 1E 85 FB CA
8FF0- BD 00 02 85 FA A0 00 B1
8FF8- FA 8D 56 8A 8A 48 A9 00
9000- 20 07 8D 20 4E 8F 68 AA
9008- CA D0 DD 60 20 F4 8A F0
9010- 52 A9 00 20 07 8D 20 88
9018- 8D 20 C6 8D AD 56 8A 20
9020- 21 8D 20 AB 8D 20 52 8D
9028- 20 64 90 A5 22 85 25 A5
9030- 20 85 24 20 80 90 F0 03
9038- 20 AD 90 E6 24 38 A5 24
9040- E5 20 C5 21 90 ED 18 65
9048- FC 85 FC 90 02 E6 FD E6
9050- 25 A5 25 C5 23 90 D8 AD
9058- 56 8A 20 21 8D 20 9E 8D
9060- 20 FD 90 60 A2 FF AD 4F
9068- 8A F0 11 20 21 BD A5 FB
9070- 9D 00 02 CA A5 FA 9D 00
9078- 02 CA D0 EA 9D 00 02 60
9080- A2 00 CA BD 00 02 F0 24
9088- 85 FB CA BD 00 02 85 FA
9090- 20 B9 8D 38 A5 24 ED 52
9098- 8A 90 E7 CD 53 8A B0 E2

90A0- A5 25 CD 54 8A 90 DB CD
90A8- 55 8A B0 D6 60 A5 25 8D
90B0- 55 8A 20 60 8D 18 A5 FA
90B8- 65 71 85 FE A5 FB 65 72
90C0- 85 FF 38 A5 24 ED 52 8A
90C8- 18 65 FE 85 FE 90 02 E6
90D0- FF A5 25 20 A7 95 38 A5
90D8- 24 E5 20 8D 51 8A A0 00
90E0- B1 FE AA AC 51 8A B1 FC
90E8- A0 00 91 FE AC 51 8A B8
90F0- 20 53 95 91 FC 8A 2C FC
90F8- 90 20 53 95 60 AD 4F 8A
9100- 48 A9 00 8D 4F 8A 20 C6
9108- 8D AD 56 8A 20 07 8D 68
9110- 8D 4F 8A 20 C6 8D A9 00
9118- 20 07 8D AD 56 8A 8D 4F
9120- 8A 20 C6 8D 60 4C 99 E1
9128- A9 00 20 07 8D AD 01 BE
9130- 20 E3 DF 85 85 84 86 20
9138- F5 E6 CA 8A 18 65 22 C5
9140- 23 B0 E2 8E 28 91 20 F5
9148- E6 CA E4 21 B0 D7 8E 29
9150- 91 20 F5 E6 8A 18 6D 29
9158- 91 CD 29 91 90 C7 C5 21
9160- F0 02 B0 C1 8D 2A 91 20
9168- F5 E6 8A F0 88 8E 2C 91
9170- E8 8A 09 C0 8D 2D 91 20
9178- B7 00 F0 09 20 BE DE 20
9180- FB E6 8E 2E 91 AD 28 91
9188- AC 2C 91 88 18 6D 2E 91
9190- B0 93 88 D0 F7 65 22 C5
9198- 23 B0 8A 90 1D CE 2B 91
91A0- AD 2B 91 C9 00 D0 1A AD
91A8- 2C 91 8D 2B 91 D0 12 AD
91B0- 2B 91 EE 2B 91 CD 2C 91
91B8- D0 07 A9 01 8D 2B 91 D0
91C0- 00 AE 2F 91 20 15 92 2C
91C8- 10 C0 2C 00 C0 10 FB A2
91D0- A0 20 15 92 AD 00 C0 2C
91D8- 10 C0 C9 95 F0 D1 C9 88
91E0- F0 BB 2C 58 8A 50 88 C9
91E8- 8A F0 C4 C9 8B F0 AE C9
91F0- 9B F0 1E C9 8D F0 0E C9
91F8- C1 90 C6 CD 2D 91 B0 C1
9200- 29 0F 8D 2B 91 CA 2B 91
9208- A9 00 20 F2 E2 20 27 EB
9210- 60 A0 00 F0 F3 AD 28 91
9218- AC 2B 91 18 88 F0 05 6D
9220- 2E 91 90 FB 20 B6 8A AC
9228- 29 91 CC 2A 91 D0 08 8A
9230- 2C 36 92 20 53 95 60 20
9238- 78 95 CB CC 2A 91 90 F7
9240- 60 8A 2C 48 92 20 AF 93
9248- 08 A2 04 20 8C 93 8D 42
9250- 92 EE 42 92 20 45 94 28
9258- EE 41 92 30 77 08 AD 56

9260- 8A 20 21 8D 90 12 20 94
9268- 94 20 C2 94 28 EE 56 8A
9270- 90 E6 20 F3 94 4C 82 93
9278- 28 B0 64 EE 56 8A 10 DD
9280- CE 41 92 10 ED 20 AF 93
9288- 08 A2 02 20 8C 93 CD 42
9290- 92 90 64 20 45 94 AD 56
9298- 8A 20 21 8D B0 53 20 9A
92A0- 94 90 48 20 60 8D A5 73
92A8- 85 FA A5 74 85 FB 20 C8
92B0- 8C 90 32 20 28 8F 20 68
92B8- 8D 20 CE 94 28 B0 12 08
92C0- EE 56 8A AD 56 8A 8D 42
92C8- 92 CD 41 92 F0 C8 90 C6
92D0- 28 C2 82 93 CE 41 92 20
92D8- F3 94 A9 05 4C 3F 95 20
92E0- 82 93 4C 3C 8B 20 82 93
92E8- 4C 31 8B 20 82 93 4C 34
92F0- 8B 20 82 93 4C 8B 20
92F8- 82 93 A2 2A 4C 12 D4 20
9300- 82 93 4C 3F 8B 20 AF 93
9308- A2 02 20 8C 93 CD 42 92
9310- 90 E5 AD 56 8A 20 21 8D
9318- 90 C5 20 45 94 20 9A 94
9320- 20 68 93 D0 DA AD 43 92
9328- 38 E9 03 8D 43 92 B0 03
9330- CE 44 92 20 77 94 20 B9
9338- 8D 20 94 94 20 C2 94 4C
9340- 82 93 20 AF 93 A2 02 20
9348- 8C 93 CD 42 92 90 8A AD
9350- 56 8A 20 21 8D 90 88 20
9358- 45 94 20 9A 94 20 68 93
9360- D0 9D 20 CE 94 4C 82 93
9368- A0 05 B9 4E 8A D1 FA D0
9370- 10 C8 B9 4E 8A F1 FA C8
9378- 38 F9 4E 8A 18 71 FA D0
9380- 00 60 20 E0 9E 20 EA A2
9388- 20 51 A8 60 8E 5F AA A9
9390- 00 8D 6C AA 8D 6D AA 8D
9398- 66 AA 8D 41 92 20 E0 9E
93A0- A9 40 20 D5 A3 20 51 A8
93A8- 20 FD 94 60 4C 34 8B 20
93B0- 95 A0 20 7B DD 20 FD E5
93B8- AA A0 00 B1 5E C9 20 D0
93C0- 06 C8 CA D0 F6 F0 E5 0A
93C8- 90 04 30 02 10 DE B1 5E
93D0- C9 2C F0 0B 09 80 99 75
93D8- AA CB CA D0 F1 F0 35 C8
93E0- CA F0 31 B1 5E C9 44 F0
93E8- 06 C9 53 F0 0E D0 BD 20
93F0- 07 94 C9 03 B0 B6 BD 68
93F8- AA 90 E4 20 07 94 C9 08
9400- B0 AA 8D 6A AA 90 D8 C8
9408- CA F0 A1 B1 5E E9 30 30
9410- 9B F0 99 60 20 B7 00 C9
9418- 2C D0 20 20 B1 00 20 E7

9420- 8D 8D 42 92 48 20 B7 00
9428- C9 2C D0 09 20 B1 00 20
9430- E7 8D 8D 42 92 68 8D 56
9438- 8A 38 60 18 A9 01 8D 56
9440- 8A 8D 42 92 60 20 FD 94
9448- EE 43 92 D0 03 EE 44 92
9450- 20 77 94 CE 42 92 F0 16
9458- 20 9A 94 38 AD 43 92 65
9460- 71 8D 43 92 AD 44 92 65
9468- 72 8D 44 92 90 E2 60 A9
9470- 00 8D 43 92 8D 44 92 AD
9478- 43 92 8D BD B5 AD 44 92
9480- 8D BE B5 A9 00 8D BF B5
9488- 8D C0 B5 A9 0A 8D BB B5
9490- 20 16 95 60 A2 03 A9 04
9498- D0 04 A2 04 A9 03 8D BB
94A0- B5 8E C1 B5 A9 00 8D C2
94A8- B5 A9 52 8D C3 B5 A9 8A
94B0- 8D C4 B5 A9 02 8D BC B5
94B8- 20 16 95 20 68 8D 20 78
94C0- 8E 60 A6 71 D0 02 C6 72
94C8- C6 71 A9 04 D0 02 A9 03
94D0- 8D BB B5 A5 71 8D C1 B5
94D8- A5 72 8D C2 B5 20 52 8D
94E0- A5 FC 8D C3 B5 A5 FD 8D
94E8- C4 B5 A9 02 8D BC B5 20
94F0- 16 95 60 AD 41 92 8D C3
94F8- B5 A9 04 D0 02 A9 03 48
9500- 20 6F 94 68 8D BB B5 A9
9508- 01 8D BC B5 20 16 95 AD
9510- C3 B5 8D 41 92 60 A2 01
9518- 20 D6 03 B0 0D AD BD B5
9520- 8D 43 92 AD BE B5 8D 44
9528- 92 60 AD C5 B5 C9 05 F0
9530- 19 C9 09 D0 0A CE 41 92
9538- 30 03 20 F3 94 A9 09 48
9540- 20 E0 9E 20 EA A2 68 4C
9548- D2 A6 A9 00 8D 41 92 20
9550- F3 94 60 84 1F 48 AD 58
9558- 8A 29 40 F0 09 AD 58 8A
9560- 10 04 68 4C 06 CF A4 1F
9568- 70 06 68 B1 28 48 50 04
9570- 68 48 91 28 68 A4 1F 60
9578- 48 B8 20 53 95 2C 58 8A
9580- 50 05 2C 1E C0 30 16 C9
9588- 20 90 10 C9 40 90 0E C9
9590- A0 90 12 C9 C0 90 06 C9
9598- E0 B0 0A 49 40 49 80 2C
95A0- A6 95 20 53 95 68 60 20
95A8- C1 FB A5 20 08 2C 58 8A
95B0- 10 01 4A 28 65 28 85 28
95B8- 60 A9 00 8D 58 8A AD B3
95C0- FB C9 06 D0 15 A9 40 8D
95C8- 58 8A AD 17 C0 30 0B AD
95D0- 1F C0 29 80 0D 58 8A 8D
95D8- 58 8A 60

Max: le moniteur étendu

Jacques Supernant

Apple II+, IIc, IIx

Ce moniteur autorise un contrôle de l'exécution des routines en langage machine.

Un mode Trace et Pas à Pas très évolués et sélectifs sont complétés par un accès direct aux registres du 6502 (ou 65C02). Il est ainsi possible de les initialiser à son gré ou à l'aide de la routine de hasard.

La gestion des fenêtres d'écran simplifie le mode trace.

Une routine permet la recherche de suites d'octets.

Un mini-assembleur très souple fait partie de Max.

Une ligne de commande peut devenir une boucle avec l'ordre JUMP

Les fichiers source sont sur la disquette.

Disquette et documentation : 150,00 F TTC franco. Bon de commande page 78.

Disquette mixte Pascal/DOS 3.3

François Sermier

Principe

Les programmes présentés ici utilisent les mêmes idées que pour la mixité ProDOS/DOS 3.3 (Pom's 18) mais sont destinés à des utilisateurs de Pascal et du Basic sous DOS 3.3 (exemple, la disquette d'accompagnement). Mais direz-vous, pourquoi pas les 3 systèmes d'exploitation à la fois sur une même disquette ? Hélas, ce serait trop beau : Pascal et ProDOS utilisent le même emplacement (bloc 2) comme clé du volume, et bien que les conceptions générales de ces blocs clés soient similaires, ils diffèrent sensiblement et sont donc incompatibles...

Rappel des principes précédemment exposés

- Initialiser une disquette Pascal à l'aide du FORMATTER UCSD
- Modifier le répertoire pour limiter le nombre de blocs Pascal
- Créer une piste CATALOG et un VTOC en piste 17 pour gérer la partie DOS 3.3.

Les deux dernières étapes seront réalisées par un programme ProDOS, INITPAS3.3. On pourrait réaliser la même chose sous DOS 3.3, mais c'est un peu plus compliqué : RWTS accède au disque par secteur, et l'interface RWTS/Basic est nettement moins agréable que MLI.

La structure d'un volume Pascal UCSD

Première remarque : la sectorisation logique de la disquette Pascal est identique à la sectorisation ProDOS. Accès par blocs et même entrelacement des blocs : ceci économise énormément de travail. Le volume Pascal débute de même par les blocs 0-1 contenant le chargeur de SYSTEM.APPLE lors du boot. Il continue par 4 blocs (2-5) du répertoire, mais, et c'est là que les choses sérieuses commencent, il n'y a pas de Volume Bit Map, et donc le bloc 6 est disponible, ainsi que les suivants (jusqu'à 279 inclus) pour les fichiers.

Conséquence : le système UCSD gère l'espace disque de manière beaucoup plus rudimentaire que ProDOS. Les fichiers doivent être stockés sur des blocs contigus (nécessité de CRUNCHer le volume de temps en temps pour regrouper les trous dans l'espace disque). Du

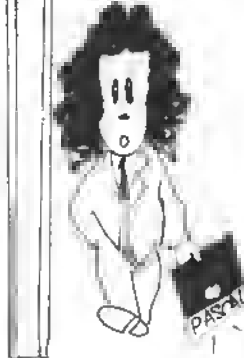
coup, l'extension de la zone Pascal au-delà de la piste 16 apporte certes des blocs supplémentaires, mais que l'on ne pourra pas agréger au reste du volume Pascal : les blocs correspondant à la piste 17 (indispensables au DOS 3.3) seront marqués défectueux pour les protéger d'un CRUNCH intempestif.

Le répertoire est composé de 78 enregistrements de 26 octets décrivant les fichiers à l'exception du premier qui décrit le volume. Du fait du

CHANGE



CHANGE



mode de gestion de l'espace disque, les fichiers figurent au répertoire dans l'ordre correspondant à leur emplacement physique sur la disquette, sans trous correspondant à d'éventuels fichiers supprimés. Dès qu'un fichier est créé (respectivement supprimé), l'enregistrement correspondant au répertoire est inséré (respectivement détruit) et les enregistrements qui le suivent sont déplacés vers le bas (respectivement vers le haut) pour compacter le répertoire. Ceci entraîne des remaniements permanents du répertoire, mais autorise une gestion simple des blocs occupés du volume par l'examen de tous les enregistrements actifs du répertoire.

Le format des enregistrements au répertoire est le suivant :

Pour un fichier :

octet numéro

0-1 numéro du bloc de début du fichier

2-3 numéro du premier bloc suivant la fin du fichier

4-5 type du fichier

6 longueur du nom du fichier

7-21 nom du fichier

22-23 nombre d'octets utilisés dans le dernier bloc du fichier

24-25 date de modification du fichier (format Pascal)

Pour le premier enregistrement décrivant le volume :

octet numéro

0-1 numéro du bloc début du volume (0 évidemment...)

2-3 numéro du bloc suivant le dernier bloc du répertoire (normalement 6)

4-5 catégorie du disque (toujours 0)

6 longueur du nom du volume

7-13 nom du volume

14-15 nombre de blocs du volume

16-17 nombre de fichiers du volume (nombre d'enregistrements actifs au répertoire)

18-19 inutilisés (0)

20-21 date du système

22-25 inutilisés.

On vérifie aisément que l'on peut reconstituer l'occupation du volume en examinant les octets 0 à 3 de tous les enregistrements du répertoire. Si vous initialisez vos disquettes en utilisant l'option 'Duplicate Directory' de la commande ZERO du FILER UCSD, vous aurez un premier enregistrement décrivant le volume identique à celui décrit ci-dessus à l'exception du champ 2-3 qui contiendra 10.

Les programmes présentés

On ne s'étonnera guère de retrouver des points communs entre les programmes proposés et INITPRO3.3 (Pom's 18). Même organisation, même appel à MLI

INITPAS3.3

INITPAS3.3 est le frère jumeau d'INITPRO3.3, toutefois, on note quelques différences.

Lignes 150-180 : si le nombre de fichiers Pascal dépasse 18, le répertoire est sur plusieurs blocs. Certaines des entrées sont à cheval entre les blocs, il est donc plus commode d'avoir l'intégralité du répertoire utile en mémoire.

Lignes 310-430 : inspection du répertoire. Tous les fichiers se terminant après le bloc 136 seront détruits et enlevés du répertoire (ligne 350). S'il en existe, la liste en est affichée et vous avez la possibilité d'abandonner. S'il existe une 2ème zone Pascal après la piste 17 (FLAG=1), on crée, ligne 430, un fichier fictif CATALOG3.3 de type défectueux (Bad) pour éviter qu'il soit Crunché, ce qui déplairait fortement au DOS...

Il serait possible d'être plus généreux pour les fichiers Pascal situés intégralement dans la 2ème région Pascal, mais cela imposerait des manipulations désagréables pour supprimer les fichiers à cheval sur la piste 17 et insérer un nouveau fichier CATALOG3.3, le tout en préservant l'ordre et la continuité du répertoire.

Lignes 440-490 : on réécrit le tout.

DUMPDIRPAS

Le programme DUMPDIRPAS vous montre comment on peut exploiter sous ProDOS le répertoire d'une disquette Pascal. Il est suivi par un exemple d'exécution que l'on pourra comparer à ce que fournit pour la même disquette le FILER UCSD.

L'option imprimante est prévue pour une imprimante permettant d'écrire en mode ultracondensé (17 caractères par pouce) pour caser un fichier par ligne. L'imprimante utilisée ici est une Image Writer, les codes de contrôle (ESC Q) sont situés ligne 170.

Le Dump hexa est réalisé en utilisant la routine moniteur PRBYTE (\$FDDB) appelée par une partie de langage machine implanté en 800 (\$320) par la ligne 3113-3114 (LDA #XX, JSR PRBYTE, JSR OUTSP, RTS), l'octet à sortir est POKé en 801 avant l'appel.

MAP

Enfin, un utilitaire permettant d'examiner des disquettes quelconques, sauf CP/M, (non protégées). Un même programme ProDOS identifie n'importe quelle disquette DOS, ProDOS ou Pascal. Si la disquette résulte d'une hybridation antérieure, cela vous est signalé en vous demandant laquelle des moitiés vous désirez examiner; néanmoins si vous avez créé une disquette mixte ProDOS/DOS sur laquelle la partie ProDOS n'atteint pas la piste 17, vous ne pourrez pas en afficher le VTOC car il faudrait pour cela aller lire un bloc situé à la fin du volume (I/O ERROR). D'ailleurs, vous n'aurez pas non plus le VBM ProDOS sauf si vous avez opportunément sacrifié les lignes 3140-3150.

Le programme vous demande dans quel lecteur est située la disquette. Ne vous étonnez pas s'il vous propose un numéro de lecteur allant de 0 à 3 :

- le numéro 3 correspond au disque virtuel /RAM (si vous avez une carte 80 colonnes étendue)
- le numéro 0 correspond à l'occupation de la mémoire principale par le programme SYSTEM du moment, c'est-à-dire dans notre cas, évidemment BASIC.SYSTEM. La table des secteurs occupés est tenue par MLI en page globale en \$BF58 - \$BF6F. Il me semble d'ailleurs qu'il y a un léger bug, car les pages mémoires \$BA à \$BD sont marquées inutilisées alors qu'elles contiennent des données de BI... Si bug il y a, on peut le corriger aisément en modifiant BASIC.SYSTEM : à l'adresse \$2048 en remplaçant LDA #\$C3 par LDA #\$FF ou encore NOP NOP, puisqu'alors l'accumulateur contient déjà \$FF.

Ayant choisi votre lecteur, le programme analyse alors l'état d'occupation de l'objet étudié en affichant l'image du VTOC (ou VBM, ou répertoire ou Memory Bit Map suivant l'OS).

L'utilité de MAP est double :

- avant hybridation, pour déterminer à partir de quelle piste on peut planter la région DOS 3.3
- après, pour les inquiets, pour vérifier que tout s'est bien passé.

L'analyse du VTOC 3.3 prend quelque temps : il faut analyser tous les bits de 70 octets, ce que le Basic ne fait pas très aisément. Si l'on s'impatiente, on peut remplacer tout le bloc 800-950 par l'appel d'une routine en langage machine (exemple, Disk Map de K.Prouty dans All About DOS).

La discrimination des systèmes d'exploitation se fait à l'aide des octets 2-3-4 du bloc 2 (c'est-à-dire piste/secteur DOS : 0/B). On vérifie en effet qu'ils contiennent dans les différents cas les valeurs suivantes (où une valeur non nulle est notée X).

Octet	2	3	4
DOS (Esclave)	0	0	0
DOS (Maître)	X	X	X
ProDOS	X	0	X
Pascal	X	0	0

Le mot 2-3 contient le chaînage du répertoire (ProDOS), des blocs occupés (Pascal). L'octet 4 contient la longueur du nom de volume (ProDOS), le type du disque Pascal (toujours 0). Les secteurs 0/A et 0/B contiennent le code permettant au DOS de se reloger (master) ou rien du tout (slave).

En utilisant les fonctions logiques, on discrimine à l'aide de la fonction : (oct2 AND NOT oct3)*(oct4 + 1), où NOT X=0 et NOT 0=1. Cette fonction vaut 1 pour ProDOS, 2 pour Pascal et 0 dans tous les autres cas.

Par ailleurs, on regarde si le contenu du secteur \$11/0 peut être un VTOC en testant l'octet de version du DOS : s'il vaut 3, on considère que la piste \$11 contient un catalogue DOS 3.3 normal (DOS=1). Ceci permet de reconnaître les disquettes mixtes pour lesquelles TYPE est différent de 0 alors qu'elles ont un répertoire 3.3 en piste \$11.

Booter ?

Pourquoi pas ? Pour booter en Pascal, il suffit de transférer SYSTEM.APPLE et SYSTEM.PASCAL sur notre disquette. On réduit ainsi, de plus de la moitié, la 1ère zone Pascal.

Simuler un boot DOS est également réalisable. En s'inspirant de la méthode décrite pour un boot DOS sous ProDOS, il faut réaliser une copie disque du DOS situé en mémoire et le transférer sur une disquette Pascal puis le charger en Pascal. Encore une vingtaine de blocs en plus sur le disque; au total il ne reste plus qu'une trentaine de blocs disponibles dans la 1ère zone Pascal.

Est-ce bien utile ? Il me semble que l'intérêt du 'mixage' réside surtout dans l'utilisation de DATA DISKS compatibles entre plusieurs systèmes d'exploitation. Cependant, il y a bien des gens (Ron De Groat dans All About Pascal) pour réaliser pire : faire coexister en mémoire et interagir le système Pascal avec le système Basic+DOS !

Bibliographie

Aux ouvrages déjà cités dans l'article précédent, on ajoutera :
All About Pascal (Call A.P.P.L.E. in Depth n[2])
All About DOS (Call A.P.P.L.E. in Depth n[3])
Le Système Pascal UCSD de Thierry Chamoret

Programme MAP

```

10 OO = 256:U = 0:F = 0:U$ = "":F$ = "":B$ = "X"
20 DEF FN W(X) = PEEK (X) + OO * PEEK (X + 1)
30 DIM T$(15),TY$(3),UN(3):TY$(0) = "DOS3.3":TY$(1) =
  "ProDOS":TY$(2) = "Pascal":TY$(3) = "Mémoire
  Vive":UN(1) = 96:UN(2) = 224:UN(3) = 176
40 GOSUB 3000
50 IF NOT DOS OR NOT TYPE THEN GOTO 90
60 : PRINT : PRINT "Map de quelle partie
  (" :TY$(0) = "0":TY$(TY) = "1" ?":GET R$: PRINT R$
70 R = VAL (R$): IF NOT NOT R < > R THEN VTAB
  PEEK (37): GOTO 60
80 TY = TY * R
90 HOME : PRINT "Volume " :TY$(TY) = "":
100 ON TYPE + 1 GOSUB 800,200,400,170
110 PRINT : PRINT "-:disponible": TAB( 15): " ":occupé"
115 IF TYPE = 3 THEN X = F : F = U : U = X
120 PRINT F: " disponibles": TAB( 13):U: " occupés"
130 END
150 X = PEEK (R)
152 FOR J = 0 TO 7
154 :OC$(I,J + D) = F$
156 :Y = INT (X / 2)
158 : IF X = 2 * Y THEN OC$(I,J + D) = U$:U = U + 1
160 :X = Y
162 NEXT
164 RETURN
170 PRINT : PRINT " :": FOR I = 0 TO 15: PRINT T$(I):
  NEXT
180 GOSUB 250
190 RETURN
200 NB = FN W(BU + 41)
210 R = BU + 4: GOSUB 600:VOL$ = N$ + "": GOSUB 650
240 POKE BLOC, PEEK (BU + 39): POKE BLOC + 1,
  PEEK (BU + 40): CALL MLI
245 PRINT
250 FOR B = 0 TO NB / 8 - 1
255 : IF TY = 3 AND B = 2 * INT (B / 2) THEN PRINT :
  PRINT "$":T$(B / 2):"0":260 :X = PEEK (BU + B)
270 : FOR I = 1 TO 8
280 : : IF X > 127 THEN X = X - 128: PRINT F$:F = F + 1:
  GOTO 300
290 : : PRINT U$:U = U + 1
300 : :X = X * 2
310 NEXT I,B: PRINT
330 RETURN
400 NB = FN W(BU + 14):NF = FN W(BU + 16):KEEP = 0
410 R = BU + 6: GOSUB 600:VOL$ = N$ + "": GOSUB 650
420 B0 = 2 + INT (26 * (NF + 1) / 512): IF B0 = 2 THEN
  GOTO 470
430 FOR I = 3 TO B0
440 : POKE PA + 3, FN HI(BU) + 2 * (I - 2): POKE
  BLOC,I: CALLMLI
450 NEXT
460 POKE PA + 3, FN HI(BU)
470 FOR I = 0 TO NF
480 :R = BU + I * 26:BB = FN W(R):BH = FN W(R + 2)
490 :BU$ = U$: IF FN W(R + 4) = 1 THEN BU$ = B$
500 : IF BB = KEEP THEN GOTO 520
510 : FOR J = KEEP TO BB - 1: PRINT F$:F = F + 1:
  NEXT
520 : FOR J = BB TO BH - 1: PRINT BU$:U = U + 1: NEXT

```

```

530 KEEP = BH
540 NEXT
550 IF BH < NB THEN FOR I = BH TO NB - 1: PRINT F$:F
  = F + 1: NEXT
560 RETURN
600 L = PEEK (R):ST = INT (L / 16):L = L - 16 * ST:N$ = "":
  IF ST = 15 THEN N$ = "7"
610 FOR J = 1 TO L:N$ = N$ + CHR$ ( PEEK (R + J)):
  NEXT
620 RETURN
650 PRINT VOL$:" blocs":NB: IF NB < > 280 THEN
  PRINT " non standard"
660 PRINT : FOR I = 0 TO 9: PRINT I: SPC( 7): NEXT
670 RETURN
800 POKE BLOC,136: CALL MLI
810 VOL = PEEK (BU + 6):NB = PEEK (BU + 52)
820 PRINT "n":VOL:" pistes":NB: IF NB < > 35 THEN
  PRINT " non standard"
830 DIM OC$(NB - 1,15): GOSUB 940
840 FOR I = 0 TO NB - 1
850 :D = 8:R = BU + 56 + 4 * I: GOSUB 150
860 :D = 0:R = R + 1: GOSUB 150
870 NEXT
880 FOR J = 0 TO 15
890 : PRINT : PRINT T$(J):
900 : FOR I = 0 TO NB - 1: PRINT OC$(I,J): NEXT
910 : PRINT T$(J):
920 NEXT
930 F = 16 * NB - U
940 PRINT : PRINT " :": FOR I = 0 TO NB - 1: PRINT T$(I -
  16 * INT (I / 16)): NEXT
950 RETURN
3000 REM ----- Initialisation Appel MLI -----
3040 MLI = 768:BUFFER = 8192:UNIT = 224: REM Drive
  1:96 --- Drive 2:224
3050 PA = 784:BLOC = PA + 4
3060 CODE = MLI + 3:LIT = 128:ECRIT = 129
3070 FOR I = 0 TO 21: READ X: POKE MLI + I,X: NEXT
3080 DATA 32,0,191,128,16,3,144,6,32,139,190
  ,32,9,190,96,0,3,224,0,32,2,0
3090 FOR I = 0 TO 15: READ X$:T$(I) = X$: NEXT
3100 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
3120 REM ----- Identification Disquette -----
3130 HOME : PRINT "Drive (0-3)?": GET R$: PRINT R$:
  OK"
3132 R = VAL (R$): IF R < 0 OR R > 3 THEN VTAB PEEK
  (37): GOTO
3130
3133 IF R = 0 THEN TYPE = 3:BU = 48984:NB = 192:U$ =
  "":F$ = "": GOTO 3180
3134 POKE PA + 1,UN(R)
3135 IF R = 3 THEN GOTO 3160
3140 POKE BLOC,136: CALL MLI
3150 DOS = ( PEEK (BU + 3) = 3)
3160 POKE BLOC,2: CALL MLI
3170 TYPE = ( PEEK (BU + 2) AND NOT PEEK (BU + 3)) *
  ( NOT PEEK (BU + 4) + 1)
3175 TYPE = TYPE * (R < > 3) + (R = 3)
3180 RETURN

```

Programme INITPAS3.3

```

10 HOME :OO = 256:FF = OO - 1
20 DEF FN HI(X) = INT (X / OO): DEF FN LO(X) = X - OO

```

```

* INT (X / OO)
30 DEF FN W(X) = PEEK (X) + OO * PEEK (X + 1)
40 DEF FN PISTE(X) = BU + 56 + 4 * X
50 DIM T$(7),TY$(2):TY$(0) = "DOS3.3":TY$(1) =
  "ProDOS":TY$(2) = "Pascal"
60 GOSUB 3000: REM Initialisation
70 IF TYPE < > 2 THEN PRINT : PRINT "Ce n'est pas une
  disquette Pascal": CALL BELL: GOSUB 3120: GOTO
  70
100 REM ----- Directory Pascal -----
102 REM
104 REM ----- Lecture
110 POKE CODE,LIT: POKE BLOC,2: CALL MLI
120 B1 = FN W(BU + 2):NB = FN W(BU + 14):NF = FN
  W(BU + 16)
130 B0 = 2 + INT (26 * (NF + 1) / 512):PM = INT ((NB - 1) /
  8)
140 R = BU: GOSUB 900:VOL$ = N$ + ":"
150 IF B0 = 2 THEN GOTO 190
160 FOR I = 3 TO B0
170 :: POKE PA + 3, FN HI(BU) + 2 * (I - 2): POKE BLOC,I:
  CALL MLI
180 NEXT
190 PRINT : PRINT "Vous allez transformer ";VOL$
200 PRINT " en mixte Pascal-DOS 3.3"
210 PRINT : PRINT "1 ère piste DOS 3.3 en n°: 17": HTAB
  27: INPUT R$:P1 = VAL (R$): IF R$ = "" THEN P1 =
  17
220 VTAB PEEK (37): HTAB 28: PRINT P1
230 IF P1 < 0 OR P1 > 34 THEN CALL BELL: PRINT "n°
  piste hors limite !": GOTO 210
240 P2 = 34: IF PM > P2 THEN P2 = PM
250 NB = 8 * P1:IH = NF
260 IF B1 < 6 THEN TR = 1: GOTO
  280
270 PRINT : PRINT "Tronquer le Directory Pascal ?N":
  HTAB 31: GET R$: PRINT R$: IF R$ = "O" THEN TR =
  1: IF IH > 38 THEN IH = 38
280 PRINT : PRINT "Nbe Secteurs Directory DOS 3.3:
  15": HTAB 32: INPUT R$:NSC = VAL (R$): IF R$ = ""
  THEN NSC = 15
290 VTAB PEEK (37): HTAB 33: PRINT NSC: PRINT
300 IF NF = 0 THEN GOTO 440
309 REM
310 REM ----- Inspection
320 BM = 136:ERR = 0:FLAG = 1: IF NB < = BM THEN BM
  = NB:FLAG = 0
330 FOR I = 1 TO NF
340 ::R = BU + 26 * I:BH = FN W(R + 2)
350 :: IF (BH < = BM) AND (I < = IH) THEN GOTO 390
360 :: IF ERR THEN GOTO 380
370 ::ERR = 1:IH = I - 1: PRINT : INVERSE : PRINT
  "ATTENTION": NORMAL : CALL BELL: PRINT "
  Vous allez détruire les fichiers": PRINT
380 :: GOSUB 900: PRINT N$: SPC( 16 - L):T$
390 NEXT
400 IF ERR THEN PRINT : PRINT "Voulez-vous continuer
  ?N": HTAB 24: GET R$: PRINT R$: IF R$ < > "O"
  THEN END
410 IF NOT FLAG THEN GOTO 440
420 IH = IH + 1:R = BU + 26 * IH: POKE R,136: POKE R +
  3090 POKE MLI + 4, FN LO(PA): POKE MLI + 5, FN
  HI(PA)
3100 POKE PA,3: POKE PA + 1,UNIT: POKE PA + 5,0

```

```

3110 POKE PA + 2, FN LO(BUFFER): POKE PA + 3, FN
  HI(BUFFER)
3111 REM
3112 FOR I = 0 TO 7: READ T$(I): NEXT :BAD$ =
  "CATALOG3.3"
3114 DATA ???? ,Bad ,Code,Text,Info,Data,Graf,Foto
3120 REM ----- Identification Disquette -----
3130 PRINT "Disquette -> Drive 2. OK?": GET R$:
  PRINT "OK"
3140 POKE BLOC,136: CALL MLI
3150 DOS = ( PEEK (BU + 3) = 3)
3160 POKE BLOC,2: CALL MLI
3170 TYPE = ( PEEK (BU + 2) AND NOT PEEK (BU + 3)) *
  ( NOT PEEK (BU + 4) + 1)
3180 RETURN

  2,144: POKE R + 3,0: POKE R + 4,1
430 POKE R + 6, LEN (BAD$): FOR J = 1 TO LEN (BAD$):
  POKE R + 6 + J, ASC ( MID$ (BAD$,J,1)): NEXT
439 REM
440 REM ----- Mise à Jour
450 POKE BU + 14, FN LO(NB): POKE BU + 15, FN
  HI(NB): POKE BU + 16,IH: IF TR THEN POKE BU +
  2,4
460 POKE CODE,ECRIT
470 FOR I = 2 TO B0
480 :: POKE PA + 3, FN HI(BU) + 2 * (I - 2): POKE BLOC,I:
  CALL MLI
490 NEXT
500 REM ----- Directory DOS 3.3 -----
502 REM
504 REM ----- Ecriture secteurs Directory
510 FOR I = 0 TO 511: POKE BU + I,0: NEXT
520 POKE CODE,ECRIT
530 POKE BU + 257,17: POKE BU + 258,14: POKE
  BLOC,143: CALL MLI
540 IMIN = 1:OC = 0:F = 0
550 IF NS < 15 THEN IMIN = 7 - INT ((NS - 1) / 2):OC =
  16382:F = 12288
560 FOR I = 6 TO IMIN STEP - 1
570 ::OC = OC - F:F = F / 4
580 ::FIN = (I < = IMIN) AND (NSC < 15)
590 :: POKE BU + 1,17: POKE BU + 257,17 * NOT FIN
600 :: POKE BU + 2,2 * I: POKE BU + 258,(2 * I - 1) * NOT
  FIN
610 :: POKE BLOC,143 - I: CALL MLI
620 NEXT
630 REM ----- Création VTOC
640 POKE BU + 2,15: POKE BU + 257,17: POKE BU +
  258,13
650 POKE BU + 3,3: POKE BU + 6,FF - 1
660 POKE BU + 39,122: POKE BU + 48,16: POKE BU +
  49,1
670 POKE BU + 52,P2 + 1: POKE BU + 53,16: POKE BU +
  55,1
680 FOR I = P1 TO P2
690 :: POKE FN PISTE(I),FF: POKE FN PISTE(I) + 1,FF
700 NEXT
710 POKE FN PISTE(17), FN HI(OC): POKE FN
  PISTE(17) + 1, FN LO(OC)
720 POKE BLOC,136: CALL MLI
890 END
899 REM --- champ Nom
900 T$ = T$( FN W(R + 4)):L = PEEK (R + 6):N$ = ""
910 FOR J = 1 TO L:N$ = N$ + CHR$ ( PEEK (R + 6 + J)):

```



```

NEXT
930 RETURN
3000 REM ----- Bip-Bip ProDOS I -----
3010 BELL = 816: FOR I = 0 TO 21: READ X: POKE BELL
+ I,X: NEXT
3020 DATA
162,32,169,2,32,168,252,141,48,192,169,36,32,1
68,252,141,48,192,202,208,237,96
3030 REM ----- Initialisation Appel MLI -----
3040 MLI = 768:BUFFER = 8192:UNIT = 224: REM Drive
1:96 --- Drive 2:224
3050 PA = 784:BLOC = PA + 4
3060 GODE = MLI + 3:LIT = 128:ECRIT = 129
3070 FOR I = 0 TO 21: READ X: POKE MLI + I,X: NEXT
3080 DATA
32,0,191,128,16,3,144,6,32,139,190,32,9,190,96,
0,3,224,0,32,2,0

```

Programme DUMPDIRPAS

```

10 HOME:OO = 256:FF = OO - 1:D$ = CHR$(4):ESC$ =
CHR$(27)
20 DEF FN HI(X) = INT(X/OO): DEF FN LO(X) = X - OO
* INT(X/OO)
30 DEF FN W(X) = PEEK(X) + OO * PEEK(X + 1)
40 DIM T$(7),TY$(2):TY$(0) = "DOS3.3":TY$(1) =
"ProDOS":TY$(2) = "Pascal"
50 GOSUB 3000: REM Initialisation
60 IF TYPE < > 2 THEN PRINT: PRINT "Ce n'est pas une
disquette Pascal": CALL BELL: GOSUB 3120: GOTO
60
70 PRINT "Sortie: Ecran (0) / Imprimante (1) ?0": HTAB
37: GET R$:OUT = VAL(R$): PRINT OUT: IF OUT < >
NOT NOT OUT THEN CALL BELL: GOTO 70
100 REM ----- Directory Pascal -----
110 POKE CODE,LIT
120 FOR B = 2 TO 5
130 :: POKE PA + 3, FN HI(BU) + 2 * (B - 2): POKE
BLOC,B: CALL MLI
140 NEXT
150 IF NOT OUT THEN GOTO 180
160 PRINT D$;"PR#1"
170 PRINT ESC$ + "Q"
180 NF = FN W(BU + 16)
190 R = BU: GOSUB 900:VOL$ = N$ + ":"
200 HOME: PRINT TAB(9);"Dump du répertoire de
";VOL$: PRINT
210 PRINT "début/ fin /type /": TAB(18);"nom": TAB(
28);"/ EOF / date ": PRINT
220 FOR I = 0 TO NF
230 :: R = BU + 26 * I
240 :: FOR J = 0 TO 25: POKE 801, PEEK(R + J): CALL
PRBYTE: NEXT
250 :: PRINT "; ";
260 :: J = 0: GOSUB 700
270 :: J = 2: GOSUB 700
280 :: PRINT T$(FN W(R + 4));"; ";
290 :: GOSUB 900: PRINT L: SPC(3 - LEN(STR$(
L)));N$:

```

```

300 :: IF I < > 0 THEN GOTO 330
310 :: PRINT SPC(8 - L); FN W(R + 14);"; "; FN W(R + 16);
"; FN W(R + 18);"; ";
320 :: D = FN W(R + 20): GOSUB 800: PRINT J$;" "; FN
W(R + 22);"; "; FN W(R + 24):: GOTO 350
330 :: PRINT SPC(16 - L);J = 22: GOSUB 700
340 :: D = FN W(R + 24): GOSUB 800: PRINT J$;
350 :: PRINT
360 NEXT I
370 IF OUT THEN PRINT ESC$ + "E": PRINT D$;"PR#3"
400 END
699 REM ----affichage tabulé
700 X = FN W(R + J): PRINT X: SPC(4 - LEN(STR$(
X)));
710 RETURN
799 REM ----champ Date
800 A = INT(D/512):D = D - 512 * A:A$ = RIGHT$("0" +
STR$(A),2)
810 J = INT(D/16):J$ = RIGHT$("0" + STR$(J),2)
820 M = D - 16 * J:M$ = RIGHT$("0" + STR$(M),2)
830 J$ = J$ + "-" + M$ + "-" + A$
840 RETURN
899 REM ----champ Nom
900 N$ = "":L = PEEK(R + 6)
910 FOR J = 1 TO L:N$ = N$ + CHR$(PEEK(R + 6 + J)):
NEXT
920 RETURN
3000 REM ----- Bip-Bip ProDOS I -----
3010 BELL = 816: FOR I = 0 TO 21: READ X: POKE BELL
+ I,X: NEXT
3020 DATA 162,32,169,2,32,168,252,141,48,192,
169,36,32,168,252,141,48,192,202,208,237,96
3030 REM ----- Initialisation Appel MLI -----
3040 MLI = 768:BUFFER = 8192:UNIT = 224: REM Drive
1:96 --- Drive 2:224
3050 PA = 784:BLOC = PA + 4
3060 CODE = MLI + 3:LIT = 128:ECRIT = 129
3070 FOR I = 0 TO 21: READ X: POKE MLI + I,X: NEXT
3080 DATA 32,0,191,128,16,3,144,6,32,139,190,32,
9,190,96,0,3,224,0,32,2,0
3090 POKE MLI + 4, FN LO(PA): POKE MLI + 5, FN
HI(PA)
3100 POKE PA,3: POKE PA + 1,UNIT: POKE PA + 5,0
3110 POKE PA + 2, FN LO(BUFFER): POKE PA + 3, FN
HI(BUFFER)
3111 REM
3112 PRBYTE = 800
3113 FOR I = 0 TO 8: READ X: POKE PRBYTE + I,X:
NEXT
3114 DATA 169,0,32,218,253,32,87,219,96
3115 FOR I = 0 TO 7: READ T$(I): NEXT
3116 DATA ????,Bad ,Code,Text,Info,Data,Graf,Foto
3120 REM ----- Identification Disquette -----
3130 PRINT "Disquette --> Drive 2. OK?": GET R$:
PRINT "OK"
3140 POKE BLOC,136: CALL MLI
3150 DOS = (PEEK(BU + 3) = 3)
3160 POKE BLOC,2: CALL MLI
3170 TYPE = (PEEK(BU + 2) AND NOT PEEK(BU + 3)) *
(NOT PEEK(BU + 4) + 1)
3180 RETURN

```


MOUSECAT

Patrice Neveu

Souriez donc !

Je veux dire par là : servez-vous de la souris de votre bel Apple //c ou de celle que vous pourrez, comme je l'ai fait, installer dans le slot 4 de votre Apple //e ou II+.

C'est le but du programme MOUSECAT OBJ que l'on lance par MOUSECAT START. Il remplit le rôle de tout bon programme de 'boot', à savoir un catalogue et le lancement aisé du programme désiré.

Utilisation

Lancer MOUSECAT START. Dès lors, l'utilisateur a le choix entre :

CATALOG : qui permet ensuite de choisir le catalogue des fichiers APPLESOFT, INTEGER, LANGUAGE MACHINE, et enfin un CATALOG normal avec tous les fichiers que l'on aurait habituellement.

A ce stade, une nouvelle feuille apparaît et permet de passer aux deux prochains secteurs du catalogue en cliquant '>>'. Chaque feuille de catalogue donne au maximum 14 fichiers, sachant qu'il y a 7 fichiers par secteur de catalogue. Il est ensuite possible de cliquer '<<' afin de revenir sur les fichiers précédents.

Pour lancer un programme, allez simplement le cliquer sur le catalogue. Vous pouvez bien sûr quitter ce

catalogue pour revenir au menu du début afin, par exemple, de changer de lecteur de disquettes. Pour cela, cliquez QUIT en bas de la page puis, revenu(e) au menu, cliquez DRIVE 1/2.

Enfin, pour sortir définitivement de MOUSECAT, appuyer sur l'option SORTIR / BOOTER. Vous pourrez alors aller vers le Basic, booter un disk, ou revenir sur votre décision et donc au menu principal.

Au niveau assembleur

Le programme *MOUSECAT* a été écrit avec le LISA 2.5 tandis que *MOUSECAT START* l'a été avec BIG MAC.

Explications sur la présentation

Imitation des logiciels du Macintosh

Le programme utilise les interruptions que la carte de la souris envoie au 6502 (lignes 68 à 123 et 480 à 550).

Cette technique a été expliquée dans le manuel vendu avec la carte (je crois, toutefois, que dans celui de la souris pour l'Apple //c, les explica-

tions ne portent que sur MOUSEPAINT) et bien sûr plus récemment dans le Pom's 15.

Depuis la ligne 125 à 246, le programme analyse ce qu'on lui demande d'exécuter. Puis, après avoir affiché un nouveau menu, il saute aux lignes 612 - 663 puis en 869 - 1023.

Les lignes 254 à 479 et 664 à 819 sont réservées aux différents sous-programmes nécessaires au bon déroulement du programme. C'est là que se situe, entre autres, CHROUT qui réalise la sortie HR des caractères ASCII dont la table est à partir de 1060. Ceux-ci sont définis ainsi :

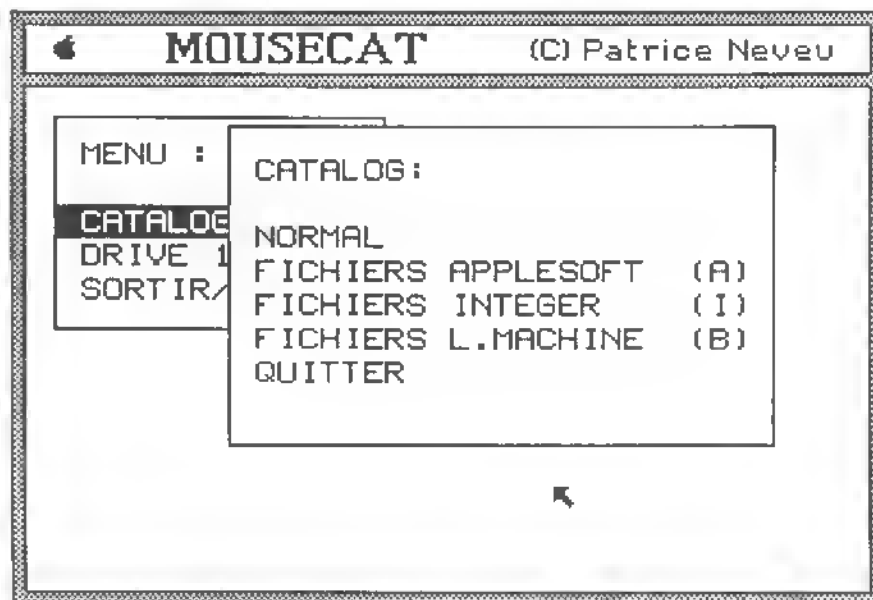
Par exemple, pour un C :

8	4	2	1	8	4	2	1	
1	1	0	0	0	0	1	1	=\$C3
1	0	1	1	1	1	0	1	=\$BD
1	1	1	1	1	1	0	1	=\$FD
1	1	1	1	1	1	0	1	=\$FD
1	1	1	1	1	1	0	1	=\$FD
1	0	1	1	1	1	0	1	=\$BD
1	1	0	0	0	0	1	1	=\$C3

C'est ici aussi qu'est la routine qui décortique le catalogue lu par RWTS. Les explications précises peuvent être trouvées dans le manuel du DOS, vendu avec l'Apple (pages 89 à 93 et pages 122 à 124).

D'autre part, on peut signaler que si MOUSECAT ne fonctionne pas en ProDOS, il fonctionne parfaitement bien en DOS 3.3, ZDOS et D-DOS. Voilà, j'espère que ceci vous séduira et permettra aux lecteurs de Pom's de mieux comprendre la structure du catalogue qu'ils sollicitent tous les jours mais qui reste souvent mystérieux, et qu'il permettra une certaine démythification de la beauté des programmes de Lisa, Macintosh, et plus récemment, de MOUSEPAINT, JANE et autres.

N.D.L.R. : l'importance des fichiers sources et objets ne nous permet malheureusement pas de les publier. Ils figurent bien entendu sur la disquette d'accompagnement.



Bibliographie

Alexandre Duback

Destination Aventure Apple //e, //plus, et //c de Delton Horn, Editions du PSI - 240 pages - 140 FF. Encore une traduction...

La façon dont on programme un jeu d'aventure est expliquée en détail autour d'un premier jeu. C'est clair et détaillé, mais pas forcément performant au niveau de la programmation. Ainsi, à un endroit, on teste la même chaîne sur une cinquantaine de valeurs possibles, au lieu d'effectuer une recherche dichotomique dans une table alpha; cela fait en moyenne 25 comparaisons au lieu de 5 ! Trois autres jeux sont listés en fin d'ouvrage, accompagnés d'explications et d'organigrammes.

Basic Applesoft Lexicum, Manuel de référence alphabétique Basic / Dos 3.3 / ProDOS de Paul Merry, Mnémodyne (26 rue Lamartine, 75009 Paris) - pavé - 168 FF.

Cet ouvrage date d'un an, mais nous n'avons pas encore eu l'occasion de le voir. Il ajoute à l'ancien manuel de référence, déjà couvert dans Pom's, 62 pages sur le ProDOS, bien utiles dans la mesure où la littérature relative au ProDOS est encore pauvre. Enfin, maintenant, la reliure est un peu plus solide...

Les matériels

Apple //c, guide de l'utilisateur de Thomas Blackadar, Sybex - 147 pages - 98 FF. Traduction.

Cet ouvrage est très général et apporte un début d'initiation au Basic, donnant à la fin un guide de référence du Basic limité aux seuls mots réservés introduits dans le livre (?). A consulter pour voir si on ne préfère pas lire la doc Apple et faire des économies.

Les ressources de l'Apple //c de Nicole Bréaud-Pouliquen, Editions du PSI - 106 pages - 85 FF.

Le premier ouvrage en français où l'on trouve un certain nombre de renseignements pour la programmation du //c, avec en particulier la liste des instructions du 65C02. A conseiller à tous ceux qui programment en assembleur sur le //c et, à présent, à ceux qui auront mis le chip du //c sur leur Apple II.

Macintosh : outils, progiciels, application de Xavier Gaucherand,

Edimicro - 237 pages - 148 FF.

Un des rares ouvrages généraux sur le Macintosh qui ne se contente pas de refaire le mode d'emploi de base du Mac avec ses logiciels "résidents" et Multiplan. Cinq parties : présentation du matériel, systèmes et langages, logiciels d'application, domaines d'utilisation et choix d'une configuration. Les analyses de points forts et faibles des logiciels sont assez bien vues. Un livre intéressant.

Logiciels et langages

dBase II - Applications de Christophe Stehly, Sybex - 268 pages - 148 FF.

Le livre a été réalisé en utilisant la version 2.4 francisée de dBase II. Une première partie illustre les possibilités du programme en gestion de fichiers; la programmation de dBase II étant couverte dans une seconde partie, à l'aide d'exemples de gestion variés. Un dernier chapitre bien intéressant explique comment il faut s'y prendre pour bien constituer ses fichiers. Ouvrage clair et bien présenté.

Le secrétariat par traitement de texte de Ilya Virgatchik, Marabout - 287 pages - 160 FF.

Nous sommes toujours dans les produits sous CP/M; bien que cela ne soit pas apparent dans le titre. Il s'agit d'un manuel pratique et complet de Wordstar.

Macintosh Microsoft Basic de Rick Dayton, Prentice Hall International - 276 pages grand format, en anglais - \$ 23.35.

Après une description sans grand intérêt du Macintosh, on trouve dans cet ouvrage... la documentation du Basic Microsoft 1.0 ! J'exagère un peu car la documentation fournie avec l'interpréteur est plus complète. Donc, si vous avez obtenu votre Basic par la voie normale, abstenez-vous d'acheter cet ouvrage qui, par contre, pourra rendre service aux pirates.

Programmez votre Macintosh de Alain Andrieux et Christophe Droulers, McGraw-Hill - 250 pages - 120 FF.

Peut être le seul ouvrage indispensable aux utilisateurs du Mac, du moins à ceux qui programment ou comp-

tent programmer leur machine. Il ne s'agit pas d'un manuel de référence du programmeur, loin de là, mais la lecture de cet ouvrage permet d'acquiescer une vue d'ensemble du "système Mac", à savoir : la gestion de la mémoire, l'interface utilisateur, etc... et surtout la fabuleuse ROM de 64Ko. Passionnant.

Macintosh, quels logiciels ? de Pierre Courbier, ETSF - 143 pages

Démonstration, de rapide à très schématique, d'un certain nombre de logiciels pour Macintosh. Un fourre-tout bien rédigé, mais dont l'utilité ne nous semble pas évidente. Réalisé en ImageWriter, mais en qualité courrier...

Multiplan et Chart sur Macintosh de Xavier Bouilloux, Edimicro - 235 pages - 159 FF.

Cela m'énerve toujours de voir un livre sorti sur ImageWriter sans même utiliser la qualité courrier. Certes, cela gagne du temps, mais quand même... La première partie, guide de l'utilisateur pour le livre, Macintosh, Multiplan et Chart, est très succinte; à peine 30 pages pour l'ensemble des deux logiciels. Ce n'est pas comme cela qu'on apprendra à les utiliser. La seconde partie comporte 10 exemples, dont 5 avec passages au graphique. L'intérêt principal de l'ouvrage tient à l'illustration de l'utilisation de Chart, sur lequel il existe moins d'ouvrages. Nous aurions préféré voir un livre uniquement sur Chart, avec un chapitre sur la communication avec Multiplan.

Multiplan sur Macintosh de Goulven Habasque, Sybex - 233 pages - 148 FF.

La première partie concerne l'initiation à Multiplan sur Macintosh, avec seulement 53 pages très aérées; ce n'est donc pas pour vous initier à Multiplan que nous vous conseillons cet ouvrage; l'initiation n'est pas assez approfondie à notre avis. Ce qui est intéressant, c'est la suite; 27 petits tableaux: d'analyse numérique (6), de géométrie (6), de statistiques (6), d'économie (5), et de gestion (4). Nous vous conseillons de consulter ce livre avant de l'acheter, pour voir si les exemples vous intéressent.

Micro-informations

Jean-Michel Gourévitch

Le mois passé, à Cupertino, on ne parlait plus que licenciements (1500 sont prévus) et baisse de la progression des ventes. Face au marasme du marché de la micro-informatique, Apple a donc du fermer trois de ses six usines, et mettre à pied un cinquième de ses effectifs.

Ces mesures drastiques ont été accompagnées d'une réorganisation interne spectaculaire. Point principal : Steve Jobs, le fondateur, a été contraint de "prendre du champ". Jean Louis Gassée, le PDG d'Apple France, s'est installé à Cupertino pour s'occuper du marketing. Del Yocam, qui couvrait la division Apple II, est devenu directeur de tous les produits. Après Steve Wozniak, qui avait quitté Apple en février, voici donc Steve Jobs sur la touche.

Ces mesures pourraient bien déboucher sur un sursaut de vitalité, et se traduire par des améliorations du Macintosh, ainsi qu'une nouvelle version de l'Apple II.

C'est tout au moins l'analyse exprimée dans un article particulièrement documenté du très sérieux Wall Street Journal, paru à la fin du mois de juin.

Dans cet article, on apprend notamment qu'Apple a décidé de jouer la partie en collaborant avec les fabricants d'accessoires. Apple a informé les fabricants de sa décision de modifier le système d'exploitation du Macintosh, afin de faciliter la gestion d'un disque dur. Jusqu'alors, l'utilisation des disques durs avec le Mac posait un certain nombre de problèmes tenant d'une part à l'incapacité du Finder à gérer plus de 134 icônes, et d'autre part à la connexion obligatoire par une des entrées séries. Le seul disque dur qui parvenait à surmonter ces handicaps était l'Hyperdrive de General Computer, installé à l'intérieur même du Macintosh. Le montage de l'Hyperdrive avait un inconvénient majeur : il fallait ouvrir le Mac, et cela annulait la garantie. Dorénavant, ce ne sera plus vrai : Apple a modifié ses conditions de garantie pour permettre l'installation d'un HyperDrive. Le Wall Street Journal pense même qu'Apple pourrait décider de vendre directement l'HyperDrive. On pensait qu'Apple allait introduire à l'automne un "Big

Mac" doté d'un nouveau système d'exploitation, et nanti d'une mémoire vive d'un mega-octet et d'un disque dur externe de 20 à 40 Mega-octets. Peut être sera-ce finalement un Mac "gonflé" et équipé d'un HyperDrive. Un système d'exploitation plus rapide, davantage de mémoire et un disque dur, voici de quoi changer la vie de tous les utilisateurs du Macintosh. A quel prix ? Là dessus, c'est encore le mystère.

Un Apple II plus puissant

Dans le même article, il est fait état d'une nouvelle version de l'Apple II, qui sortirait dans les douze prochains mois, et serait doté d'un microprocesseur plus rapide ainsi que de disquettes d'une capacité plus importante. Ce portait ressemble énormément à l'Apple II équipé d'un microprocesseur 16 bits dont les lecteurs de ces Micro-informations ont déjà pu lire la description. Quant au lecteur de disquettes miracle, il s'agit d'un modèle fabriqué par Sony sous le nom "Liron", qui utilise les mêmes disquettes de trois pouces et demi déjà employées sur le Macintosh. Seule différence : ce lecteur est double face. Sa capacité : 400K par face, soit 800K au total. Pas mal non, comparé aux 143K actuels des lecteurs de l'Apple II ? Ce lecteur miracle confèrera même au II une capacité de stockage plus importante que celle du Macintosh. Il ne serait, dans ces conditions, pas impossible que l'Apple II devienne un jour un concurrent sérieux du Mac. Ce que Steve Jobs voulait, dit-on, éviter à tout prix.

Baisses de prix ?

Aux Etats Unis, on commence à trouver l'Apple IIc pour environ 600 dollars. En France, Apple a commencé à vendre une promotion composée d'un Apple IIc, d'AppleWorks, et d'un sac de transport pour 9990F. Officiellement, cette "affaire" est réservée aux étudiants. Mais qui ne connaît pas un étudiant ?

Des fenêtres pour le II

Les programmeurs se sont en tous cas remis à travailler sur ce vieil

Apple II, en exploitant toutes les ressources de la souris, des fenêtres et des menus déroulants. Certains programmes entrevus ressemblent donc, d'ores et déjà, furieusement à ceux du Macintosh. On peut en attendre toute une vague pour le Sicob, ou pour Noël. Une nouvelle version d'Appleworks avec souris et menus déroulants serait ainsi sur le métier. En France, Version Soft préparerait un traitement de texte graphique très semblable à MacWrite. Chez d'autres concepteurs de logiciels, on pense tableurs, ou réalisation de graphiques.

Un super tableur

En attendant, le programme vedette de l'été pour l'Apple II n'utilise ni souris, ni graphiques, ni menus déroulants. Pire ! ce n'est ni un intégré, ni un programme révolutionnaire façon "Think Tank". Il s'agit d'un banal tableur. Un tableur ? Quelle idée, direz-vous, après Visicalc qui fit la fortune d'Apple, et Multiplan qui fit celle de Microsoft. Eh bien, **Supercalc3a** de **Sorcim** semble actuellement tout simplement le plus puissant des tableurs pour le II. Jugez plutôt : 254 lignes, 63 colonnes et 15000 cases disponibles. Une gestion très fine de la mémoire n'utilisant cette denrée précieuse que pour les cases contenant des données, une vitesse de traitement quatre fois supérieure à celle du tableur d'Appleworks (sur un II équipé d'une carte accélérateur, Supercalc tournerait deux fois plus vite que sur un IBM PC), 50 fonctions intégrées, aussi bien trigonométriques que financières. Et la possibilité, en prime, de traduire les données en graphiques et en couleurs, ou de les recevoir ou les envoyer par modem. Voici un programme (dans l'esprit de Lotus 1-2-3, bien que ce ne soit pas un intégré) qui transforme un tranquille Apple II en super IBM, pour 195 dollars.

Retombées d'Apple Expo

A la mi-juin, les visiteurs d'Apple Expo ont pu admirer (outre Super-

Calc) une fournée de logiciels nouveaux. Parmi eux : **ClickWorks** de **Contrôle X**, un gestionnaire de fichiers graphiques souris avec tableur intégré (2350 F), **Version Com**, de **Version Soft**, un programme de communications avec souris et menus déroulants, **Varicalc de Del-taSoft**, créé par un pilote de ligne pour réaliser ses plans de vols, c'est devenu un tableur multi-entrées affichant sous forme de courbes l'interaction de plusieurs paramètres.

Les possesseurs de //e peuvent, eux, disposer d'une nouvelle carte d'extension couleur avec 64K supplémentaire : c'est la **Féline**, qui remplace l'ancienne carte EVE, et offre des capacités graphiques très étendues. Les graphiques de Version Calc en double haute résolution graphique prennent ainsi, en couleurs, une allure étonnante. 2520F.

Pendant ce temps, aux Etats Unis, un programme utilitaire connu de tous les utilisateurs d'Apple n'en finit pas de progresser : c'est le copieur **Copy II Plus** de **Central Point Software**. On en est à la version 5.0 et il est devenu automatique : on tape le nom du programme à copier, et le paramétrage s'effectue automatiquement. 40 dollars. **Street Electronics**, le grand spécialiste des interfaces, a conçu le **LiveWire**. C'est un câble de liaison avec l'imprimante "intelligent". Il comporte une carte horloge/calendrier incorporée dans son connecteur, et un adaptateur convertissant la sortie série en sortie parallèle (permettant d'utiliser d'autres imprimantes que l'ImageWriter avec un Apple //). Moins de 100 dollars.

Du Hard pour le Mac

En attendant que le Mac devienne un jour un système "ouvert", le voici doté d'une impressionnante gamme d'accessoires et d'extensions. Le mois dernier, Pom's vous parlait d'une possibilité d'extension à 1 Mega. Cette modification est désormais possible en France, grâce à **MacMega** de **P-Ingénérie**.

Cette filiale de **Polygone informatique** se propose de transformer un Macintosh 128k ou 512K en Macintosh de 1024K. La mémoire centrale élargie étant directement adressable, tous les logiciels peuvent l'utiliser. Avec le switcher, c'est un rêve, on peut faire tourner plusieurs applications simultanément, sans se trouver à court de mémoire. Avec un disque virtuel (un programme simulant un lecteur de disque dans la mémoire

de Macintosh), tel que **Speedy**, conçu aussi par **P-Ingénérie** les programmes tournent avec une rapidité spectaculaire. Cette modification coûterait 15000 francs pour un Macintosh 128K, et 10000 pour un Mac 512K. L'opération est effectuée en cinq jours.

Attention cependant : en l'état actuel des conditions de vente, cette greffe de mémoire annule la garantie du constructeur.

Autres accessoires intéressants pour le Mac

Le **Port Power d'Assimilation**. Cet adaptateur qui se branche sur la sortie imprimante permet d'utiliser le Mac avec des accessoires fonctionnant en "parallèle", et notamment tous les périphériques IBM. Mais une sortie située sur l'adaptateur permet de conserver l'usage de la sortie série. Série ou parallèle, plus besoin de choisir. 79 dollars.

Le **Mac Nifty Switch** du **Kette Group** permet, lui, de brancher deux périphériques sur le même port, et de choisir entre les deux par un bouton (prix 35 dollars), le **Mac Nifty Stereo Music System** transforme le son monophonique du Mac en son pseudo-stéréophonique (prix 80 dollars, ampli et haut parleurs compris), enfin le **MacNifty Joystick Converter** permet d'utiliser un joystick en guise de souris (45 dollars). Utiliser un autre accessoire que la souris, c'est décidément un rêve partagé. La **MacTablet** de **Summagraphics** remplace la souris par une tablette et un stylo optique. Un must pour les dessinateurs et les illustrateurs, que la souris rebutait.

Moins intéressant

Dès qu'un appareil, quel qu'il soit, a du succès, on voit fleurir autour de lui une foule d'accessoires plus ou moins étranges; le Mac n'échappe pas à la règle. **MouseMat** est un tapis destiné à recevoir les déplacements de la souris, avec mise à la masse du Mac pour évacuer les charges électrostatiques. 540 FF. **Mouse-Medic** est un kit de nettoyage, toujours pour la souris, qui comporte un pinceau soufflant, une bombe d'air comprimé, une petite bouteille de liquide de nettoyage "spécial", et... un chiffon ! Le tout pour 300 FF. **MouseHouse** est une petite boîte en plastique qui vient se coller sur un

flan du Mac, afin de recevoir la souris lorsqu'elle n'est pas utilisée.

Très joli... 120 FF.

Le principal avantage de ces accessoires, croyez-en notre expérience, c'est qu'on peut très bien s'en passer !

Logiciels, du sérieux

Macintosh a suscité une avalanche de programmes de gestion de données, et ces programmes ont à leur tour enfanté d'une nouvelle génération de bases de données, le plus souvent relationnelles, dotées parfois de macros-commandes et transformables en fonction des besoins de l'utilisateur final par une société de service informatique. Parmi ceux-ci : **Quatrième dimension d'ACI**, **Omnis 3**, diffusé par KA, et **Mac-Base 500** de **Contrôle X**.

Le Macintosh avait déjà un tableur extra : **Multiplan** de Microsoft. Il fallait l'astuce de Microsoft pour le transformer en super-tableur; c'est **Excel**. Sortie prévue à l'automne. Tableur et Graphiques intégrés, 256 rangées, 16384 colonnes, 42 types de graphiques multi-fenêtres. On l'a compris, c'est du sérieux. Bizarrement, ce super-tableur est apparemment plus proche de Lotus 1-2-3 que ne l'est Jazz. Excel deviendra à coup sûr un des best-sellers des logiciels pour Macintosh.

Pour dessiner

Tandis que sort la version 1.8 de **MacDraw**, voici un autre programme qui lui ressemble bigrement, mais qui comporte de notables améliorations, c'est **MacDraft** de **IDD**. Possibilité de "zoomer" sur une partie du dessin, pour y ajouter de fins détails, d'opérer un panoramique, pour voir l'ensemble, 16 échelles, possibilité d'opérer des rotations d'objets par incréments d'un degré, ou de distribuer ces objets le long d'une ligne, possibilité d'avoir 4 documents séparés sur l'écran simultanément, et d'y transférer des éléments, possibilité d'y coller des images de MacPaint, ligne fléchée pouvant indiquer automatiquement les dimensions d'un objet. Après un super tableur, voici un super Mac-Draw. Pour désigner un dossier ou une application, **Icon Switcher** de **PBI Software** permet de dessiner et de personnaliser les icônes (Prix 19 dollars). Et pour ceux qui manquent de talent ou d'imagination, il existe déjà des icônes toutes dessinées : **Icon Libraries** (prix 19 dollars).

Pour imprimer

Il est loin le temps où les utilisateurs de Macintosh étaient strictement limités à l'utilisation de l'ImageWriter d'Apple. **SoftStyle** propose des "drivers" permettant d'utiliser avec le Mac une imprimante Epson (**Eps-tart** : 45 dollars), une ThinkJet Hewlett Packard (**Jetstart** : 45 dollars), une LaserJet d'Hewlett Packard (**Laserstart** : 95 dollars).

Pour les fanas des statistiques

Statview de **Brainpower** est un programme de statistiques interactif, permettant la vision simultanée des données et des analyses. Prix 195 dollars. **StatFast** de **Statsoft** permet toutes corrélations, Anova, Régressions multiples, etc... et l'obtention de graphiques. Prix : 99 dollars. **StatWorks** de **Heyden & Son** dispose des mêmes fonctions, d'une précision de 19 chiffres après la virgule, intègre aussi un modèle graphique permettant d'obtenir des histogrammes en trois dimensions, et d'utiliser 50000 points de données : 100 dollars. Enfin, **ClickGraph** de **Contrôle X**, plus simple et moins scientifique, est la partie graphique

de CX MacBase enrichie de possibilités de Zoom, et capable de récupérer des données de Chart ou Multiplan. Prix : 1200F.

Enfin des simulations.

L'un des plus éblouissants jeux de simulation s'appelle **Gato** de **Spectrum Holobyte**. Il permet à l'utilisateur de simuler le pilotage d'un sous marin pendant la seconde guerre mondiale. Jusqu'à présent, ce jeu réalisé avec des sons et des images tridimensionnelles n'existait que pour l'IBM PC. Le voici transposé (et amélioré) sur le Macintosh. Images superbes et simulation hyper réaliste. Prix: 59 dollars. A quand un simulateur de vol sur Macintosh ?

Adresses

Sorcim/IUS Micro Software
2195 Fortune Drive, San Jose, CA 95131

Contrôle X
94, rue Lauriston 75116 Paris

Version Soft
66, rue Castagnary 75015 Paris

Central Point Software
9700 SW Capitol Hwy#100 Portland OR 97219

Street Electronics Corporation
1140 Mark Avenue Carpinteria. CA 93013

P-Ingenierie
226, bd Raspail 75014 Paris

Kette Group
13895 Industrial Park Blvd · Minneapolis MN 55441

Summagraphics Corp.
777 State Street Extension P.O. Box 781 Fairfield CT 06430

IDD
1975 Willow Pass Road Suite 8, Concord CA 94520

PBI Software
1155 B H Chess Drive Foster City CA 94404

SoftStyle
7192 Kalaniannaole Hwy Suite 205 Honolulu Hawaii 96825

Brainpower
24009 Ventura Bd, Calabasas, CA 91302

Statsoft
2831 East 10th St. Suite 3, Tulsa OK 74104

Heyden & Son
247S 41st St, Philadelphia PA 19104

Spectrum Holobyte
1050 Walnut, Suite 325, Boulder, CO 80302

Courrier des Lecteurs

Olivier Herz

J'attire votre attention sur la routine de chargement rapide des fichiers binaires (Mr Tran-Van, Pom's N° 4 : en mode direct elle est parfaite, mais en mode programme, la commande "& suivi d'une variable" ne fonctionne que si elle est précédée d'un "& suivi d'un nom réel de programme". Si vous en doutez, prenez le programme "& BLOAD DEMO" sur la disquette n°, supprimez la première partie de la ligne 20: faites RUN; résultat... : I/O ERROR.

D'autre part, dans le POM'S n° 16, rubrique Trucs et astuces, le programme Basic du "Euréka il y a un PEEK pour savoir si un point graphique est allumé", ne fonctionne pas. C'est probablement une faute de frappe quelque part (à la 1ère ligne A = PEEK (Y/64 est déjà quelque

chose d'impossible : imaginez que Y = 12, par exemple, et calculez le PEEK...). Solution ?

M. Tomeno Paris

Il y a effectivement un bug sur le & BLOAD, mais pas tout-à-fait de la façon dont vous le décrivez : & BLOAD A\$ "plante" avec un I/O ERROR, quand il suit un accès disque autre qu'un & BLOAD "TOTO" (RUN, BLOAD, CATALOG ou VERIFY, par exemple).

Après étude, on s'aperçoit que le numéro de volume est indiqué au DOS dans le cas du & BLOAD "TOTO" (lignes 85 et 86 du source), mais pas dans le cas du & BLOAD A\$. Cela explique que le bug n'ait pas lieu lorsqu'un & BLOAD A\$ suit un & BLOAD "TOTO".

Pour y remédier, il suffit de décaler les lignes 85 et 86 avant la ligne 80, et je propose le patch suivant :
BLOAD CHARGE.OBJET,A\$9000
9035 : A0 00 8C EB B7 20 B7 00
C9 22 D0 23 A9 06 85 C1
BSAVE CHARGE.OBJ,
A\$9000,L\$22B
(bien noter D0 23 et non D0 28 comme dans l'original).

Il y a effectivement deux coquilles dans le "trucs et astuces" du n° 16. Il faut lire :

1ère ligne : A = PEEK (Y/64)
9ème ligne : P = (P-INT(P))) = .5

Les EPROMs des cartes imprimantes Epson portant les références APLB, APLD et APLE comportent des bugs qui, pour les deux premières, empê-

chent l'assemblage sur papier avec MERLIN et, pour la dernière, perturbe les tabulations. Les personnes intéressées peuvent prendre contact avec moi ; j'ai en effet reprogrammé des EPROM sans ces inconvénients et dotées de nouvelles fonctions : formattage numérique avec la virgule et non le point, génération très simple de nouveaux caractères, dump écran texte.

Yvan Koenig, Mosaïque Gerbino, rue du stade, 06620 Valauris.

Je vous soumetts ces quelques problèmes :

Comment charger ICARE à partir d'un programme sans en interrompre le déroulement ?

car :
10 PRINT CHR\$(4); "BLOAD
ICARE,A\$8000"
20 CALL 32768
30 PRINT "ICARE chargé"
s'arrête à l'instruction 20

Je possède la carte 80 colonnes de CHAT MAUVE. En utilisant ICARE en mode 80 colonnes, plus aucun accès disque n'est possible. Le DOS semble déconnecté. Seul un RESET permet de reprendre la main; comment cela se fait-il ?

La commande MAXFILES ne marche plus après le chargement d'ICARE en 80 ou en 40 colonnes ! Comment puis-je faire passer "MAXFILES 5" tout en utilisant ICARE ?

En regardant attentivement le source, j'ai remarqué en ligne 875 le message & PROUT; de même, en 792, en pressant l'OA', on obtient & PROUT. Cette commande n'engendrant pas de message d'erreur, je pense qu'elle fait quelque chose, mais quoi ?

M. Amsaleg Lamorlaye

ICARE redonne la main au Basic après son chargement. Pour que cela ne se produise pas, je vois deux solutions :

- patcher le programme; mais cela oblige à trouver dans le source, puis l'objet, la modification à faire, ce qui est un peu lourd;
- créer et lancer un fichier EXEC qui contient :
BLOAD ICARE 3.1,A\$8000
CALL 32768
RUN (application)

ICARE n'a pas été écrit pour fonctionner en 80 colonnes, aussi ne vous étonnez pas si son fonctionnement en 80 colonnes pose des problèmes. Toutefois, je dois vous signaler que cela ne déconnecte en principe pas le DOS, sauf peut-être si l'on fait un & HGR PRINT.

MAXFILES vaut 3 par défaut lors du boot, ce qui place la HIMEM à

\$9600. Mettre MAXFILES à 4 descend la HIMEM à \$93AD, ce qui cause un conflit avec la partie "table" (page 35) de ICARE. Par conséquent, ICARE ne supporte pas que l'on donne à MAXFILES une valeur supérieure ou égale à 4.

Après examen du listing source, je pense que & PROUT "TOTO", A\$,B\$, "TITI" envoie une chaîne de caractères à l'imprimante sans passer par un PR#1; mais, à l'instar du & HARDCOPY, cela ne marche que sur une Epson équipée d'une interface Epson.

Transférer un écran 80 colonnes pour pouvoir le récupérer intégralement, après avoir utilisé l'écran à d'autres fins : cela doit être anodin ! Après avoir passé un jour, et surtout une partie de la nuit suivante, j'en conclus que ce n'est pas aussi anodin qu'il n'y paraît.

En deux mots :

Je transfère le 1/2 écran en mémoire principale de 1024 à 2048, vers la mémoire auxiliaire de 10240 à 11264.

Idem pour le 1/2 écran en mémoire auxiliaire que je transfère de 1024 à 2048 vers la mémoire principale de 10240 à 11264, au moyen de la routine AUXMOVE.

J'efface partiellement l'écran.

Je transfère dans l'autre sens, les 2 1/2 écrans sauvegardés en mémoire principale et mémoire auxiliaire de 1024 à 11264 vers 1024 à 2048 respectivement en mémoire principale et mémoire auxiliaire, hélas... un seul revient.

M. De Crecy. Versailles

Votre programme fonctionne parfaitement bien... à condition de ne pas être en 80 colonnes (mais on ne peut alors vérifier que la seconde moitié revient correctement).

En effet, la mise en 80 colonnes met le commutateur 80STORE en position ON (par l'équivalent d'un POKE -16383,0), ce qui a pour conséquence que AUXMOVE ne fonctionne pas correctement avec la partie de la mémoire correspondante à la page texte (\$400-\$7FF), car AUXMOVE utilise les commutateurs RAMRD et RAMWRT sur lesquels 80STORE a priorité. Par conséquent, vous devez mettre 80STORE en position OFF (par l'équivalent d'un POKE -16384,0) avant d'appeler AUXMOVE. Par exemple :

```
LDA #$0
STA $C000; met 80STORE off
SEC ou CLC selon la direction :
principale (-) auxiliaire
JSR $C311; appel de AUXMOVE
LDA #$0
```

STA \$C001; remet 80STORE on RTS

Peut-on transformer un fichier avec par exemple : A\$(80,80),B\$(200,1) en fichier binaire, pour accélérer le chargement (environ 2 minutes avec DOS 3.3) et qu'il puisse opérer avec un programme Basic utilisant des A\$(I,L) ? Mes essais ont été infructueux.

M. Boyer Chatillon

En ce qui concerne le problème de la transformation d'un tableau de chaînes en fichier binaire, il faut, pour le faire, écrire une routine en assembleur, relativement complexe (à la différence d'une routine identique pour les tableaux de variables réelles), car, si le descriptif du tableau des chaînes est une table placée à un endroit connu de la mémoire, les chaînes en revanche se trouvent dispersées dans la zone comprise sous la HIMEM.

La seule réalisation à ma connaissance sur ce sujet, est celle de Nibble "AMPER-SPEED" (dans Nibble Express Volume III, que vous pourrez trouver chez votre revendeur, SIVEA par exemple). Le gain de vitesse est d'un facteur trois seulement, car la routine ne crée pas de fichier binaire, mais accélère les opérations sur les fichiers texte en court-circuitant les commandes du DOS.

Utilisation Gesmask modifié (POM'S n° 14) sous PRODOS : je n'arrive pas à le faire fonctionner, il doit sans doute y avoir des changements d'adresses, donc de lignes de programme, mais lesquels ? Mon niveau actuel ne me permet pas d'y répondre. De plus, je suppose que BASICIUM ne doit pas fonctionner sous PRODOS.

M. Swiderski Bourgoin-Jallieu

En ce qui concerne Gesmask modifié sous ProDOS, il faut faire les modifications habituelles pour que le programme Applesoft marche sous ProDOS :

- il faut remplacer à la ligne 20 :
D1\$=CHR\$(13)+D\$ par
D1\$=D\$.
- faire un BLOAD d'une image texte (\$400-\$7FF) donne un "NO BUFFERS AVAILABLE" car ProDOS n'aime pas charger un fichier à cet endroit.

Pour y remédier, il suffit de faire la même chose que sous DOS (Pom's n° 9, page 27; errata Pom's n°10 page 73).

Le BASICIUM fonctionne parfaitement sous ProDOS, après quelques opérations indiquées dans son manuel d'utilisation.

pom's

Disquettes

HAIFA source	(cf. Pom's n° 5)	à 55,00 F
H-BASIC	(cf. Pom's n° 8)	à 150,00 F
MUSIC	(cf. Pom's n° 10)	à 80,00 F
DBSTAG	(cf. Pom's n° 11)	à 450,00 F
JEUX A	(cf. Pom's n° 12)	à 80,00 F
JEUX B	(cf. Pom's n° 12)	à 80,00 F
BASICIUM	(cf. Pom's n° 13)	à 150,00 F
E.P.E.	(cf. Pom's n° 15)	à 150,00 F
PASCAL	(cf. Pom's n° 15)	à 80,00 F
MAX (Moniteur étendu)	(cf. Pom's n° 18)	à 150,00 F
DOMINOS	(cf. Pom's n° 19)	à 80,00 F

Recueils

N°1, recueil des revues 1 à 4	à 140,00 F
Disquettes d'accompagnement 1 à 4	à 150,00 F
N°2, recueil des revues 5 à 8	à 140,00 F
Disquettes d'accompagnement 5 à 8	à 190,00 F

Anciens numéros

Revue	4 7 8	à 35,00 F
Revue	9 10 11 12 13 14 15 16 17 18 19	à 40,00 F
Disquettes Apple II, //e, //c	1/2 3 4 5 6 7 8 9 10 11 12	à 55,00 F
	13 14 15 16 17 18 19		
Disquettes Macintosh			
	14/15/16 groupées	à 150,00 F
	17 18 19	à 80,00 F

Abonnements

Pour 6 numéros à partir du n°

Abonnement à la revue	à 200,00 F
Abonnement aux disquettes Apple II, //e, //c	à 280,00 F
Abonnement aux disquettes Macintosh	à 400,00 F

Total TTC :

Supplément avion hors CEE : 15,00F par numéro et/ou disquette :

Montant du règlement :

Envoyez ce bon et votre règlement à : EDITIONS MEV, 64 rue des Chantiers 78000 VERSAILLES

Nom :

Adresse :

Les résultats du sondage

Fichier Edition Présentation Rangement Lecteurs Sondage Pourcentages									
Lecteurs : 823		Basic : 96.20 %		Débutant : 13.57 %		Apple /// : 1.12 %		Revue : 28.45 %	
Hommes : 97.58 %		Pascal : 31.78 %		Intermédi. : 33.46 %		Apple //c : 12.45 %		Kiosque : 18.32 %	
Femmes : 2.42 %		Forth : 4.46 %		A. averti : 45.72 %		Apple][+ : 25.09 %		Salons : 6.90 %	
- de 15 : 1.49 %		Logo : 13.20 %		Haut niv. : 7.25 %		Apple //e : 58.92 %		Boutique : 19.40 %	
15 à 19 : 6.88 %		6502 : 36.06 %		Disq. dur : 2.04 %		Mac 128K : 8.18 %		Amateur : 26.94 %	
20 à 24 : 12.27 %		65002 : 3.72 %		Table Tr. : 1.30 %		Mac 512K : 5.39 %		80 colon. : 26.35 %	
25 à 34 : 29.00 %		68000 : 1.67 %		Pom's lus : 13.13		Lisa : 0.93 %		80 c. ét. : 36.47 %	
35 à 45 : 31.41 %		Lisp : 3.16 %		N. person. : 3.27		Revue : 18.92 %		Couleur : 31.29 %	
+ de 45 : 18.96 %		Cobol : 4.46 %		Niveau Génér. : 4.24		Kiosque : 31.08 %		CP/M : 15.30 %	
Travail : 53.35 %		Fortran : 6.88 %		Présentation : 3.50		Salons : 6.76 %		Mao + A[[: 29.73 %	
Gestion p. : 44.05 %		LSE : 0.19 %		Intérêt progr. : 3.78		Boutique : 27.03 %		Impriman. : 97.30 %	
Jouer : 52.42 %		Prolog : 1.12 %		Originalité pr. : 3.86		Amateur : 16.22 %		Niveau g. : 4.49	
Plaisir pr. : 86.06 %		C : 0.56 %		Clarté artiol. : 3.19		Banos d'ess. : 3.23		Présentat. : 3.97	
Heures/se		Domicile : 74.72 %		Intérêt pédag. : 3.47		Micro-infos : 3.65		Intérêt pr. : 3.90	
		Dom. + T. : 21.00 %				Disquettes : 3.86		Origin. pr. : 3.83	
								Clarté art : 3.42	
								Intérêt pè. : 3.54	
								Banos d'es : 3.20	
								Micro-inf. : 3.03	
								Disquettes : 3.95	

Heures par semaine :

Moins de 2 heures :	3.35 %
De 3 à 8 heures :	41.08 %
De 9 à 16 heures :	34.76 %
Plus de 16 heures :	20.82 %

Voici les résultats du sondage proposé dans Pom's n° 17 et auquel 823 lecteurs ont bien voulu répondre.

Il est difficile de dire si l'échantillon des personnes qui ont répondu est représentatif ou non du lectorat complet de notre revue. Cependant, vous remarquerez que le nombre de réponses est assez grand pour que les résultats soient significatifs, même s'ils ne sont pas parfaitement représentatifs.

Nous constatons, tout d'abord, et cela confirme nos observations faites à partir du courrier des lecteurs, le faible pourcentage de lectrices; c'est dommage ! Ensuite, sachez que vous êtes plus de la moitié, sur la totalité des lecteurs qui ont participé au son-

dage, à passer plus de neuf heures à câliner votre matériel. Bravo.

Pour tous ceux qui croyaient que Pom's était l'exclusivité de quelques jeunes "bidouilleurs de génie", désolés de les décevoir; la revue est lue par des personnes de tous âges.

En moyenne, chaque lecteur possède 13 numéros de notre revue, vous êtes donc un grand nombre à avoir la collection complète; vous êtes sur la bonne voie, nous l'apprécions.

Au niveau des langages, Basic est en tête, c'est normal... Néanmoins, dans les divers assembleurs, nous obtenons un pourcentage de plus de 40 %, excellent résultat. Pascal assure une bonne prestation, suivi de près par la performance de Logo. LSE ? nous en rions encore !

Le tableau ci-joint, obtenu à l'aide d'un programme "maison", n'indique bien sûr que le résultat de données "discrètes". Vos appréciations ont été analysées "manuellement" et appellent quelques commentaires. Tout d'abord, vous êtes dans l'ensemble satisfaits de la revue; nous

allons donc continuer dans le même sens, tout en améliorant certains points comme, par exemple, la présentation. Vous pouvez d'ailleurs constater que, dans ce numéro, les listings des programmes sont beaucoup plus lisibles que par le passé, et nous essaierons de faire fois la prochaine mieux ! Les utilisateurs d'Apple II comprennent, acceptent et, dans la plupart des cas, apprécient la présence du "cahier Mac". Les utilisateurs du Macintosh sont satisfaits du "cahier", bien que regrettant le nombre de pages relativement limité consacré à leur appareil. Toutefois, nous pensons qu'une dizaine de pages de renseignements utiles sont préférables à cinquante pages de... n'importe quoi. Si l'on veut prendre un exemple, le listing de "Lucy", publié dans ce numéro, en dit beaucoup plus sur le fonctionnement de la machine que la totalité de ce que l'on a pu voir jusqu'à présent dans la presse spécialisée. A noter que près de 30 % des utilisateurs du Mac utilisent aussi un Apple II.

Voulez-vous en avoir plus pour votre argent dans le domaine des câbles informatiques ?

Si vous êtes utilisateur, acheteur ou prescripteur, envoyez-nous dès aujourd'hui votre nom sur papier à en-tête. Vous n'aurez qu'à vous en féliciter.

Prière de renvoyer à Belden Electronics GmbH (Service 4040 Neuss-Uedesheim (Allemagne de l'Ouest).

PF Fuggerstrasse 2,

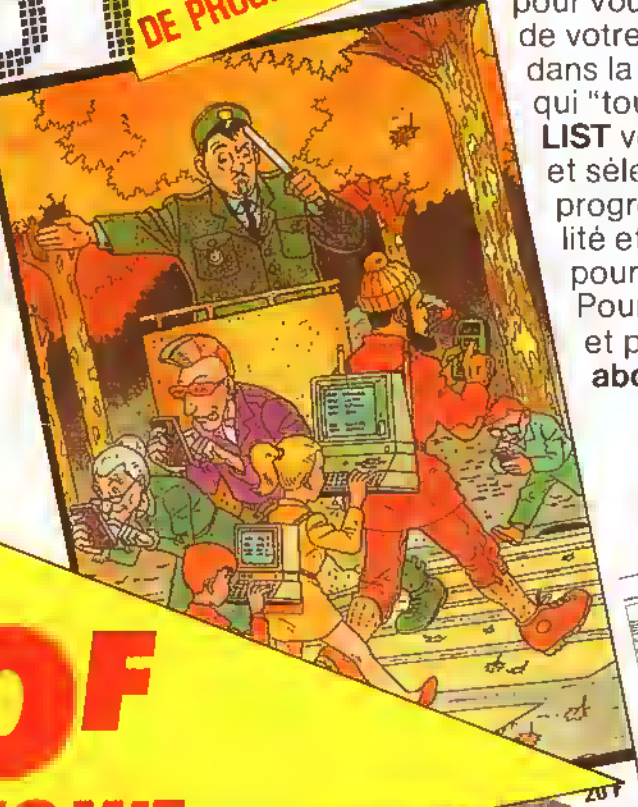
LIST

LE JOURNAL DES AMATEURS DE PROGRAMMATION

LE JOURNAL DES AMATEURS DE PROGRAMMATION

Dans
chaque numéro :

- les "trucs" de votre machine.
- des programmes commentés.
- Basic, Forth, Pascal et les autres langages.
- des problèmes de programmation.
- les nouveautés : matériels, logiciels, livres.

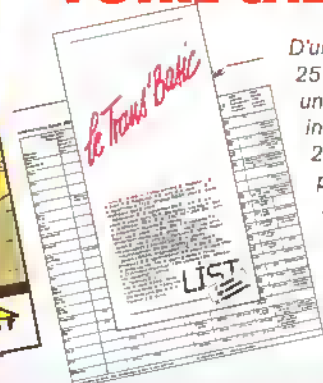


40F

D'ECONOMIE

Si programmer un ordinateur est devenu pour vous un loisir, un plaisir... une passion, sachez que **LIST** a été créé pour vous. **LIST** vous aide à tirer davantage de votre matériel, à vous perfectionner dans la conception des programmes qui "tourneront" sur votre machine. **LIST** vous initie aux langages informatiques et sélectionne les meilleurs livres pour progresser. **LIST** vous informe de l'actualité et vous fournit trucs, astuces et idées pour mieux programmer... Pour être sûr de ne rater aucun numéro et pour recevoir **LIST** chez vous, **abonnez-vous !**

VOTRE CADEAU !



D'un format pratique (11,5 x 25 cm), le **TRANS'BASIC** est une table de conversion des instructions **BASIC** des 21 micro-ordinateurs les plus courants. Ce livret de 24 pages vous permettra d'adapter sur votre ordinateur la plupart des programmes conçus pour les autres machines.

BULLETIN D'ABONNEMENT

(à retourner à LIST - Service Abonnements - 5, place du Colonel Fabien - 75491 Paris Cedex 10)

Nom _____ Prénom _____

Adresse _____

Ville _____ Code postal | | | | | Pays _____

Veuillez m'abonner pour 10 numéros au prix avantageux de 160 F* au lieu de 200 F - prix total au numéro - et m'adresser en cadeau le **TRANS'BASIC**. Je joins mon règlement indispensable libellé à l'ordre de **LIST**.

* Belgique: 1330 FB; Suisse: 50 FS; Canada: 30 \$C; autres pays: 210 FF. Par avion: Afrique Francophone: 245 FF; Amérique, eutre Afrique, Océania: 305 FF; Asie: 355 FF. Belgique: Soumillion, 28, av. Massenet, 1190 Bruxelles. Versement Société Générale 2100405 835-38. Suisse: 19, route du Grand-Mont, CH 1052, Le Mont-sur-Lausanne, versement Caisse d'Épargne et de Crédit, 10-2418 Le Mont CH 1052, compte courant n° 650156-7. Canada: LMPI, 9345, rue de Maux, St Léonard (Québec), H1R 3H3, Canada. Autres pays: 5, place du Colonel Fabien, 75491 Paris Cedex 10.